Making Disk Failure Predictions SMARTer!

Sidi Lu
Wayne State University

Bing Luo
Wayne State University

Tirthak Patel
Northeastern University

Yongtao Yao
Wayne State University

Devesh Tiwari
Northeastern University

Weisong Shi
Wayne State University

Abstract

Disk drives are one of the most commonly replaced hardware components and continue to pose challenges for accurate failure prediction. In this work, we present analysis and findings from one of the largest disk failure prediction studies covering a total of 380,000 hard drives over a period of two months across 64 sites of a large leading data center operator. Our proposed machine learning based models predict disk failures with 0.95 F-measure and 0.95 Matthews correlation coefficient (MCC) for 10-days prediction horizon on average.

1 Introduction

Hard disk drives (HDDs) continue to be a key driving factor behind enabling modern enterprise computing and scientific discovery — residing in large-scale data centers. Unfortunately, HDDs are not only the most frequently replaced hardware components of a data center; they are also the main reason behind server failures [82]. The failure of HDDs can result in data loss, service unavailability, increases in operational cost and economic loss [42, 76]. Consequently, the storage community has invested a significant amount of effort in making disks reliable and, in particular, predicting disk failures [4, 9, 19, 23, 24, 36, 41, 51, 54, 58, 59, 85, 89, 92]. Although widely-investigated, effective hard disk failure prediction still remains challenging [83, 88] and hence, the storage community benefits from the disk reliability field-studies [8, 37, 44, 53, 55, 60, 65, 77, 83, 88]. Unfortunately, such field studies are not published often enough and are limited in sample size [8, 9, 28, 30, 37, 60, 83, 88, 89].

To bridge this gap, we perform large-scale disk failure analysis, covering 380,000 hard disks and five disk manufacturers distributed across 10,000 server racks and 64 data center sites over two months, hosted by an enterprise data center operator — one of the largest disk failure analysis studies reported in the literature [4, 9, 51, 83].

For the first time, this paper demonstrates that disk failure predictions can be made highly accurate by combining disk performance and disk location data with disk monitoring data (Self-Monitoring, Analysis, and Reporting Technology — SMART data). Traditionally, disk failure prediction works have largely focused on using SMART data for predicting disk failures — this is based on in-the-field evidence that SMART attributes (e.g., correctable errors, temperature, disk spin-up time, etc.) are correlated with the disk health and indicative of eventual failure. While this conventional wisdom holds true as shown by previous works, we found that SMART attributes do not always have the strong predictive capability of making disk failure predictions at longer prediction horizon windows for all disks (i.e, predicting disk failures a few days before the actual failure instead of a few hours). This is primarily because the value of SMART attributes often does not change frequently enough during the period leading up to the failure, and the change is often noticeable only a few hours before the actual failure, especially in hard-to-predict cases.

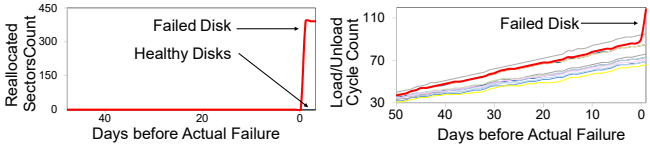

Figure 1: SMART attributes of healthy vs. failed disks prior to disk failures.

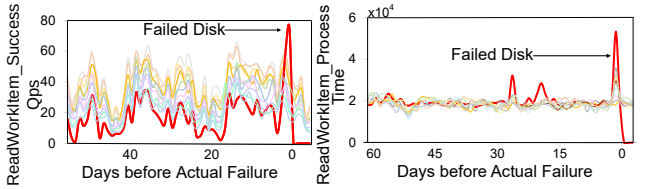

Figure 2: Performance metrics of healthy vs. failed disks prior to disk failures.

On the other hand, the value of performance metrics may exhibit more variations much before the actual drive failure. A small example is shown in Figure 1 and Figure 2. We observe that the performance metrics of failed disk drives may indeed show distinguishable behavior from healthy disks (Figure 2) while SMART attributes do not (Figure 1). In Figure 1, the SMART attributes of healthy disks show the same value or similar pattern as failed disks located on the same server until the time of disk failure. For the performance metrics shown in Figure 2, although the trends of failed disks are close to healthy disks, failed disks may report multiple sharp impulses before they actually fail. Only a subset of SMART attributes are shown in the plot, but others also show similar behavior (our methodology is

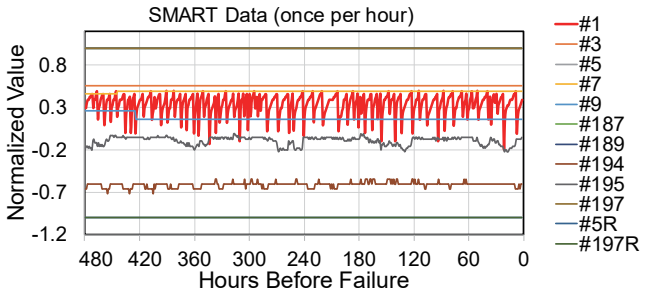

Figure 3: Values of SMART attributes before a hard disk failure, collected on an hourly basis, extracted from the open-source Baidu dataset [40]. The legend on the right shows the IDs of disk SMART attributes as defined by the industry standard [3], and "R" represents the raw value of an attribute.

covered in Section 2). We note that this example evidence does not suggest that *all* failed disk drives show variation in performance metrics leading up to the failure, or that SMART attributes do not change for any failed disks. Instead, it shows that performance metrics, when combined with a traditional approach of using SMART attributes, may be more powerful than using SMART attributes alone, especially for hard-to-predict failures.

One could argue that SMART attributes not exhibiting distinct patterns between healthy and failed disks is specific to this data center under study. To test this hypothesis, we plotted the normalized value of SMART attributes of failed and healthy disks from a publicly available disk failure dataset released by Baidu in 2016 [40]. Figure 3 shows that the normalized values of 12 SMART attributes of a randomly selected failed disk do not vary noticeably leading up to the failure — 477 hours before its actual failure. This observation is particularly notable, especially, given that the SMART attributes for this dataset are collected at much finer-granularity (one hour) as opposed to traditional per-day granularity (e.g., Backblaze public dataset [46]). Thus, SMART attributes alone may not be able to predict all disk failures.

Intuitively, the addition of performance metrics toward disk failure prediction increases the predictive power because it increases our coverage in capturing the workload characteristics accessing the storage system, beyond what SMART attributes cover. The nature of workloads running on a system often affects the failure rates of different system components, not only disks. But, it's much more challenging to obtain and incorporate workload related information due to the business-sensitive nature of data center workloads. As shown in Section 5, performance metrics can act as a good proxy for workload characteristics for disk failure prediction.

Finally, this paper shows that disk failure prediction can be further improved by incorporating the location information of disk drives in the data center — an aspect that has not been explored in the previous disk failure prediction works because typically data center logs do not include location and organization of disks by default. Intuitively, the addition of location information toward disk failure prediction increases the predictive power because it increases our coverage of the operating conditions of data center disks.

Disks in close spatial neighborhoods are more likely to be affected by the same environmental factors, such as relative humidity and temperature, which are responsible for accelerating disk component failures [55, 73]. Notably, disks with physical proximity are likely to experience similar vibration levels. Although vibration is not a part of the SMART attributes or performance metrics, it is known to affect the reliability of disk drives [56, 65]. Therefore, adding location information can capture disks operating under similar environmental or operating conditions which can experience similar failure characteristics. Our evaluation (Section 5) shows that adding location information to SMART attribute information indeed improves the failure prediction quality, although as expected, the effects are not as large as adding performance metrics to SMART.

While using the combination of SMART attributes, performance metrics, and location information is likely to improve disk failure prediction quality, the types of attributes, and the raw amount of combined information is almost unmanageable. It is unclear what attributes should be selected and how they should be used. Traditional rule-based or analytical models are not likely to exploit the hidden interactions among different attributes of the same type (e.g., SMART) and different types (e.g., performance vs. SMART). Therefore, to increase the effectiveness of our approach, we take advantage of machine learning (ML) models for leveraging such hidden interactions, as done in several previous disk failure prediction works [9, 51, 54, 65, 89].

Our core contributions are not in the development of machine learning based models, built on top of well-understood and mature models such as naive Bayes classifier (Bayes) [36], random forest (RF) [52], gradient boosted decision tree (GBDT) [29, 91], and long short-term memory networks (LSTM) [23, 38]. Instead, the core usefulness of our study is in providing actionable insights, trade-off lessons learned in applying these models, and assessment of model robustness. Additionally, we develop and evaluate a new hybrid deep neural networks model, convolutional neural network long short-term memory (CNN-LSTM) [2] for disk failure prediction that achieves close to the best prediction quality in most of the test cases.

Summary of Our Contributions:

⋆ *This paper presents findings from one of the largest disk failure prediction studies covering 380,000 hard drives over a period of two months across 64 sites of a leading data center operator.* Our disk failure prediction framework and the dataset used in this study including performance, SMART, and location attributes is hosted at `http://codegreen.cs.wayne.edu/wizard`.

⋆ *This paper provides experimental evidence to establish that performance and location attributes are effective in improving the disk failure prediction quality.* We show that, as expected, machine learning based models can be useful in predicting disk failures. But, as we discover and discuss in this paper, there are several trade-offs in the model selection. We also understand, discuss, and explain the limitations of these models. This paper provides details of an experimental and evaluation methodology for effective disk failure prediction.

⋆ *Overall, our evaluation shows that no single machine learning model is a winner across all scenarios, although CNN-LSTM is fairly effective across different situations. We achieve up to 0.95 F-measure [66] and 0.95 MCC (Matthews correlation coefficient) [10, 35, 43, 71] score for a 10-day lead-time prediction horizon (Refer to Section 4 for the definitions of F-measure and MCC).* We show that combining SMART attributes, performance metrics, and location records enables us to do disk failure prediction with long lead-times, although the prediction quality changes with the lead time window size.

2 Background and Methodology

This study covers the disk and server data measured and collected at a large data center. Over, the dataset spans over 64 data center sites, 10,000 server racks and 380,000 hard disks for roughly 70 days. This corresponds to roughly 2.6 million device hours [4, 9, 51, 83]. We note that during this period, the data center housed more than two million hard disks, but not all of them are included in our study because we only focus on those disks that have logged data in all three aspects: SMART, performance, and location. Collection and storage of both performance and SMART data are not enabled for all disks due to performance overhead and business-sensitivity concerns.

Next, we assess the types of disk events recorded at the data center sites and describe our definition of disk failure. Then, we discuss all three types of data collected and analyzed for this study: (1) disk SMART attributes (most commonly used for disk failure prediction by other studies [4, 19, 79, 87], (2) performance data, and (3) spatial location data of disks.

Table 1: SMART attributes for disk failure analysis.

ID	Attribute Names	ID	Attribute Names
1	Read Error Rate	7	Seek Error Rate
9	Power-On Hours	192	Power-off Retract Count
10	Spin Retry Count	193	Load/Unload Cycle Count
3	Spin-Up Time	194	Temperature
12	Power Cycle Count	197	Current Pending Sector Count
4	Start/Stop Count	198	Uncorrectable Sector Count
5	Reallocated Sectors Count	199	UltraDMA CRC Error Count

2.1 Definition of Disk Failure

Given the complexity of disk failures, there is no common, agreed-upon universal definition of a disk failure [53]. Latent sector errors (LSEs) are typically considered to be one of the most common disk errors which cause disk failures. However, a large-scale study of disk failures [75] shows that a small number of LSEs alone do not necessarily indicate that a disk failure has occurred or is imminent, but LSEs may cause performance degradation that could eventually lead to a "failure" — where error messages such as "the system cannot connect to the disk" or "disk operation exceeded the prescribed time limit" are treated as disk failures and warrant disk replacement. In this paper, we consider a disk to be failed when the production data center operator deems a disk necessary to be replaced. The IT operators of the production data center we study deem it appropriate for a disk to be replaced or repaired when there is a failed read/write operation and the disk cannot function properly upon restart. All other disks are considered healthy.

2.2 Disk SMART Data

SMART attributes values are produced and logged under the Self-Monitoring, Analysis and Reporting Technology (SMART) monitoring system for hard disks, which detects and reports various indicators of drive reliability [3]. The number of available SMART attributes is more than 50, but not all disks log all of the attributes at all times. For our study, we select 14 SMART attributes (Table 1) as features for our training models using the method described in Section 3. More than 97% of our disks reported these attributes, and these attributes also overlap with the widely used attributes for disk failure prediction by other studies [9, 51, 54, 65, 89]. In our study, these SMART attributes are collected continuously and reported at per-day granularity during the whole duration of the data collection period, similar to previous works [37, 54, 54]. As discussed earlier, more frequent SMART reporting did not necessarily improve the prediction quality at the start of this study and hence, once-a-day reporting was employed.

In our study, we consider two values corresponding to each SMART attribute in Table 1: (1) raw value of the attribute, and (2) normalized value of the attribute. Raw values are collected directly by the sensors or in-

Table 2: Selected disk-level performance metrics.

ID	Metrics	ID	Metrics
1	DiskStatus	7	Background_Checksum_ReadFailQps
2	IOQueueSize	8	TempFile_WriteWorkItem_SuccessQps
3	ReadSuccess_Throughput	9	TempFile_WriteSuccess_Throughput
4	ReadWorkItem_QueueTime	10	NormalFile_WriteWorkItem_SuccessQps
5	ReadWorkItem_SuccessQps	11	NormalFile_WriteWorkItem_QueueTime
6	ReadWorkItem_ProcessTime	12	NormalFile_WriteSuccess_Throughput

ternal software in disks, and their interpretation can be specific to the disk manufacturer. Normalized values are obtained by mapping the related raw value to one byte using vendor-specific methods. Higher normalized value usually indicates a healthier status, except in the case of head load and unload cycles and temperature. We note that whether a higher (or lower) raw value is better often depends on the attribute itself. For example, a higher value of "Reallocated Sectors Count" represents that more failed sectors have been found and reallocated (worse case), while a lower value of "Throughput Performance" indicates a possibility of a disk failure.

2.3 Performance Data

In our study, we measure and collect two types of performance metrics maintained by the OS kernel, i.e., disk-level performance metrics and server-level performance metrics. Disk-level performance metrics include IOQueue size, throughput, latency, the average waiting time for I/O operations, etc. Server-level performance metrics include CPU activity, page in and out activities, etc. Performance metrics are reported at per-hour granularity because we found that hourly granularity was effective in improving the prediction quality. However, the storage overhead of all performance metrics can become significant at scale and over time, and it can incur significant operational costs. Therefore, as described in Section 3, we use a simple method to down-select the number of metrics used by our ML models to manage prediction quality with low storage overhead.

2.3.1 Disk-level Performance Metrics

In our study, we measure and collect 12 disk-level performance metrics in total; all of these metrics are used in this paper. Table 2 shows the 12 metrics related to individual disks.

The distinct value of "DiskStatus" represents different disk working statuses. For example, 0, 1, 2, 4, 8, 16, 32, 64 and 128 indicate healthy, initial, busy, error, hang, only read, shutdown, repair, and complete repair states, respectively. "IOQueueSize" shows the number of items in the IO worker queue. "NormalFile/TempFile_WriteSuccess_Throughput" represents the throughput of normal/temp files successfully written to disks. "NormalFile/TempFile_WriteWorkItem_SuccessQps" and "ReadWorkItem_SuccessQps" stand for the number of normal/temp files successfully written/read by the disk per second. Similarly, "NormalFile_WriteWorkItem_QueueTime" indicates the average waiting time for disks to write. "ReadSuccess_Throughput," "ReadWorkItem_ProcessTime," and "ReadWorkItem_QueueTime" indicate the throughput, process time, and the average waiting time through the reading process of disks.

Table 3: Selected server-level performance metrics and the corresponding categories.

Categories	Metrics	Categories	Metrics
disk_util	max	udp_stat	udp_outdatagrams
tcp_segs_stat	tcp_outsegs	disk_sys_read_write	read
page_activity	page_in	net_pps_summary	net_pps_receive
disk_summary	total_disk_read	net_summary	receive_speed
disk_throughput	read	page_activity	page_out
disk_util	avg	udp_stat	udp_indatagrams
memory_summary	mem_res	disk_summary	total_disk_write
tcp_currestab	NONE	net_pps_summary	net_pps_transmit
cpu_summary	cpu_kernel	tcp_segs_stat	tcp_insegs

2.3.2 Server-level Performance Metrics

As to the server-level metrics, we have 154 metrics categorized into 54 categories; each category has a different number of metrics. We first extract the most common pairs of category-metrics and make sure that more than 97% of servers have these server-level metrics. We down-select the number of metrics to 18 that we feed to our machine learning model to manage prediction quality vs. storage overhead via a simple method described in Section 3. Table 3 lists the 18 server-level performance metrics and their corresponding categories.

"Tcp_outsegs" displays the total number of the disk storage segments that have been sent, including those on current connections but excluding those containing only retransmitted octets. Similarly, "tcp_insegs" shows the total number of disk storage segments received, and "tcp_currestab" represents the number of TCP connections for which the current state is either established or close-wait. "Udp_outdatagrams" displays the total number of the disk storage UDP datagrams that have been sent. "Page_in" represents the number of transferring data from a disk to the memory per second. Similarly, "page_out" occurs when the data is transferred from the memory to a disk. Packets per second (PPS) is a measure of throughput for network devices. Hence, "net_pps_receive" and "net_pps_transmit" indicate the rate of successfully receiving and transmitting messages over a communication channel, respectively. Note that the performance data also includes network-related (TCP/UDP) metrics some of which appear in the selected server-level performance metrics; this suggests that network- and disk- activity might be correlated and may be predictive of disk failures when combined.

2.4 Disk Spatial Location Data

As noted in Table 4, our disks are spread over more than 50 sites and 10,000 racks. All disks are directly

Table 4: Numbers of sites, rooms, racks, and servers.

	# of Sites	# of Rooms	# of Racks	# of Servers
Total	64	199	10,440	120,000

attached to a server. Each disk has four levels of location markers associated with it: site, room, rack, and server. One server may host multiple disks. Multiple servers could be on the same rack. A room has multiple racks, and a site may host several rooms. Location markers are used for both healthy and failed disks. Note that these location markers do not explicitly indicate the actual physical proximity between two disks, since the physical distance between two sites or rooms is not captured by our location coordinates, and they do not indicate the physical proximity within a room.

2.5 Other Methodological Considerations

Our disk failure prediction study is carefully designed to ensure that it is not prone to experimental pitfalls. For example, we verified that the disk failure rate is roughly similar over time across all 64 sites because if most disk failures happen during the same week it can skew the prediction quality. Similarly, we ensured that the concentration of disk failures in space is not skewed. Although failures in space have non-uniform distribution, we have verified that the density of failures in space changes over time. Our annual disk failure rate of ≈1.36% is consistent with failure rates observed at other data centers [47–49].

We note that missing SMART or performance data is a possibility and can itself be indicative of the system's health. For example, if failed disks observe a higher degree of missing data than healthy disks and the failed disks have been missing data continuously for a long period (e.g., more than the prediction horizon), then this feature alone could predict disk failure with high success rate. However, in our case, we observed that healthy and failed disks do not have an imbalance in terms of missing data. Furthermore, the length of continuous missing data is less than one day in most cases because we have multiple types of data: performance and SMART. The likelihood of missing all samples from both groups simultaneously is low — and if data is missing, it often points to an abrupt disk/server failure or other infrastructure-related issues.

We also ensure that disk failures are not concentrated on a particular manufacturer only, or limited to only old-aged drives. Although our datset has multiple vendors and drives of different ages, we verified that failure prediction does not reduce to trivially knowing the vendor name or age of the disk — although these features are used by our machine learning models to improve the quality of prediction. We explored training and building vendor-specific ML models, but we found that this leads to multiple problems: (1) overfitting to a particular vendor, (2) lack of portability across sites and vendors, (3) managing multiple models, and (4) lower prediction quality than the approach taken in this paper (normalizing the attributes across vendors and disks as discussed in Section 3).

3 Selection of SMART and Performance Attributes

In this section, we present a simple method to down-select SMART and performance metrics. These down-selected metrics are then fed to our machine learning models as input features. Unless otherwise noted, we use this method for selecting important features and use the resulting features to present evaluation results. However, one could argue that machine learning models can automatically infer important features out of all the input features. The reason for performing this step is to demonstrate that down-selecting features using a simple method does not compromise the prediction quality, as we evaluate in Section 5. The benefit of this step is the saving in storage overhead. Although our study needed to store all the features (over 100) to demonstrate the effectiveness of down-selection, in the future, data center operators can use the method to save storage space and reduce processing overhead. Since the range of values for different attributes across different disks and vendors varies widely, it is hard to perform meaningful comparisons. Thus, we pre-process the SMART and performance metrics using a min-max normalization to facilitate a fair comparison between them as per equation: $x_{norm} = (x - x_{min})/(x_{max} - x_{min})$. Here, x is the original value of a feature, x_{min} is the minimum and x_{max} is the maximum value of the feature (over all observations). We use 0 to represent the NULL value, and we label constant features as 0. Next, we leverage Youden's J index (also named as J-Index) [27, 74] for the down-selection of features.

3.1 How does J-Index (JIC) work?

After features are normalized to the scale of 0-1, we set a series of threshold candidates for each feature with a step of 0.01, starting from 0 until 1. For each threshold candidate t, we calculate the value of the corresponding J-Index [6]. We define J-Index classification (JIC) as:

$$\begin{aligned}\text{J-Index} &= \text{True Positive Rate} + \text{True Negative Rate} - 1 \\ &= \frac{\text{TP}}{\text{TP+FN}} + \frac{\text{TN}}{\text{TN+FP}} - 1\end{aligned}$$

Here T and F indicate whether the prediction result is correct; P and N denote the disk is classified as failed (positive) or healthy (negative). TP denotes the number of actually failed disks that are correctly predicted as

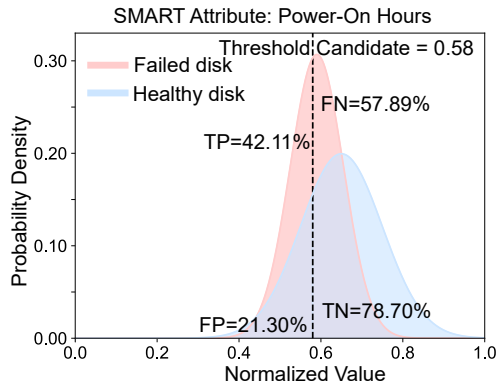

Figure 4: An example of J-Index classification (JIC): Distinguishing failed disks from healthy disks. The upper curve represents the failed disk, and the lower curve indicates the healthy disk.

failed, and TN denotes the number of healthy disks that are correctly predicted as healthy. Similarly, FP denotes the number of healthy disks that are falsely predicted as failed, and FN denotes the number of failed disks that are falsely predicted as healthy.

More specifically, suppose the input feature is Power-On Hours, and the distribution looks like Figure 4 for the current threshold candidate t (t = 0.58 as an example here). We calculate the percentage of failed disks that are distributed on the left-hand part of t, which is 42.11%, i.e., TP = 42.11%. Similarly, we have FN = 57.89%, FP = 21.30%, and TN = 78.70%. It is intuitive that we predict a disk is healthy if its value is greater than 0.58 or it is otherwise failed. We also calculate the corresponding J-Index based on the above definition. Following this method, for a specific feature, we have a series of threshold candidates and their corresponding J-Indexes. The range of J-Indexes is 0 to 1. A higher J-Index means the corresponding threshold candidate is more distinguishable to identify failed disks from healthy disks. Therefore, the threshold candidate with the highest J-Index is selected as the best (final) threshold for a feature.

Intuitively, J-Index classification is a low-overhead and practical method for IT operators to adopt and perform feature selection on their datsets.

Table 5 shows the J-Indexes (greater than 0.1) for SMART attributes. The fourth and sixth columns (yellow color) represent the percentages of disks that are smaller than the threshold, while the fifth and last columns (blue color) show the percentages of disks that are greater than the threshold. For each attribute, the first bold font indicates the true positive rate, and the second bold font denotes the true negative rate. Since failures are not always supposed to be values that are less than the threshold, i.e., there are upper-bound thresholds and lower-bound thresholds for failed disks, the

Table 5: Highest J-Indexes for SMART attributes (R represents raw value, N denotes normalized value).

ID	Threshold	J-Index	% of failed disks		% of healthy disks	
9R	0.58	0.21	**42%**	58%	21%	**79%**
9N	0.54	0.19	52%	**48%**	**72%**	28%
3R	0.72	0.18	80%	**20%**	**98%**	2%
5R	0.49	0.18	**18%**	82%	0%	**100%**
194N	0.50	0.18	**45%**	55%	27%	**73%**
194R	0.50	0.15	**96%**	4%	81%	**19%**
1R	0.38	0.14	**28%**	72%	14%	**86%**
12N	0.04	0.14	65%	**35%**	**79%**	21%
4N	0.01	0.13	64%	**36%**	**77%**	23%
5N	0.01	0.13	87%	**13%**	**100%**	0%
3N	0.59	0.13	77%	**23%**	**90%**	10%

Table 6: Highest J-Indexes for performance metrics.

ID	Threshold	J-Index	% of failed disk		% of healthy disk	
2	0.13	0.45	**100%**	0%	55%	**45%**
3	0.11	0.44	2%	**98%**	**46%**	54%
7	0.10	0.40	8%	**92%**	**48%**	52%
11	0.16	0.40	8%	**92%**	**49%**	51%
6	0.15	0.38	12%	**88%**	**50%**	50%
9	0.12	0.31	25%	**75%**	**56%**	44%
8	0.10	0.30	27%	**73%**	**56%**	44%

bold values for the true positive rate and true negative rate span multiple columns.

Similar to SMART attribute analysis, we would like to see if performance metrics could be the indicators of disk failures. Table 6 shows a part of the highest J-Indexes for performance metrics following the same formatting guide as Table 5. By employing the JIC method, we figure out a set of most informative disk-level and server-level performance metrics that are indicative of impending disk failures, i.e., we select the metrics that have the highest J-Indexes (greater than 0.1). We also present the best (final) thresholds of some of the selected metrics in Table 5 and Table 6.

Contrary to SMART attributes, performance metrics tend to have a higher true positive rate and a lower true negative rate. We observe that although a single performance metric is not perfect to distinguish failed disks from healthy disks, it has an overall higher J-Index than most of the SMART attributes based on our dataset. This indicates that performance metrics are likely to be predictive for disk failures.

Next, we show that performance metrics of failed disks may show different distinguishing patterns before failure compared to the healthy disks. Recall that there are 12 disk-level performance metrics in total. For each server that contains one or more failed disks (failed server), we extract these 12 metrics of each disk within 240 hours before disks are reported to be failed. If there is only one failed disk on a specific failed server, we keep the raw value of the failed disk (RFD) and calculate the average value of all healthy disks (AHD) for every time point. Then, we get the difference between RFD and AHD, which indicates the real-time difference between the signatures of failed disks and healthy disks on the

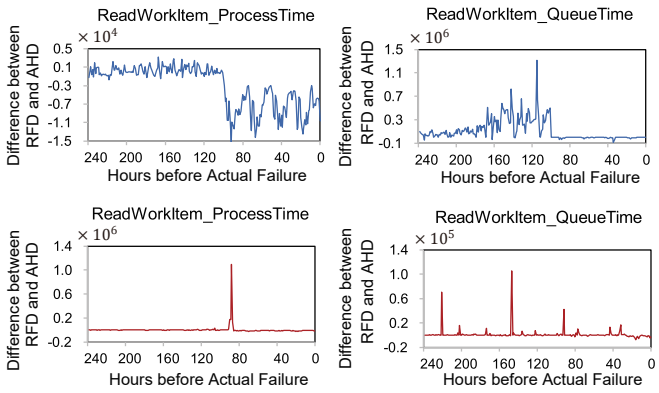

Figure 5: Different types of patterns of performance metrics observed 240 hours before disks failure.

same server. If there are N ($N \geq 2$) failed disks, then for each failed disk, we calculate the difference between RFD and AHD for every time point.

Figure 5 shows representative samples of the difference between RFD and AHD curves for different performance attributes on different servers. To reveal the patterns more intuitively, we use the raw values of metrics to calculate the difference between RFD and AHD rather than the normalized values in Figure 5. All disks on the same server have the same value of server-level performance metrics, and hence, 18 selected server-level performance metrics are not shown in the plot. The top two graphs of Figure 5 illustrate that some failed disks have a similar value to healthy disks at first, but then their behavior becomes unstable as the disk nears the impending failure. The bottom two graphs of Figure 5 show that some failed disks report a sharp impulse before they fail, as opposed to a longer erratic behavior. These sharp impulses may even repeat multiple times. We did not find such patterns for SMART attributes so far before the failure of this selected example. The diversity of patterns demonstrates that disk failure prediction using performance metrics is non-trivial.

4 ML Problem Formulation and Solution

Problem Definition. We formulate the problem of predicting disk failures as a classification problem. Specifically, we use $T = \{(\text{input}_i, \text{label}_i)\}_{i=1}^{n}$ to represent our training dataset, in which $\text{input}_i \in I$ denotes all input features. Here, $\text{label}_i \in \{0,1\}$ is a binary response variable for each disk i: 0 indicates healthy state and 1 indicates failed state. Our goal is to employ the best method to learn the function $f : I \rightarrow \{0,1\}$, which minimizes the loss function $\ell(h(\text{input}); \text{label})$, a measurement of the difference between the desired output and the actual output of the current model, such that the trained model is able to predict disk failures $(\text{label}_i = 1)$ over a specific prediction horizon with high accuracy.

More specifically, during the training process, assume we only use one attribute a as an input feature. For each disk, we have multiple readings of the attribute: a_1, a_2,..., a_n (j is the time in a_j), and we treat $\{a_1,...,a_n\}$ as a sample. Since the input of a machine learning algorithm should be a fixed length of the observation period for each sample, n should be a fixed number. Our goal is to predict disk failure in advance, so a_j in $\{a_1,...,a_n\}$ should be the value of healthy states (of the healthy disks or healthy states prior to failures), i.e., a_j in $\{a_1,...,a_n\}$ does not contain failed state data. Note that we aim to predict *if* the disk will fail and not the exactly *when* the disk will fail in the next ten days.

Effective Measurements. To evaluate the effectiveness of our prediction approaches, we use Precision, Recall, F-measure, and Matthews correlation coefficient (MCC) to measure the wellness of our prediction approaches. Precision [22] indicates the proportion of TP among all predicted failures. Recall that the true positive rate (TPR) [81] represents the proportion of TP within all actually failed disks. Since our binary classification is largely imbalanced — there are many more healthy disks than failed disks — we also use F-measure [39, 69] and MCC [10] as our evaluation metrics. F-measure is the harmonic average of precision and recall and ranges between 0 and 1 (higher is better). *We use MCC because it is a more balanced measure than F-measure, especially suitable for imbalanced data. It ranges from 1 (perfect prediction) to -1 (inverse prediction).* These metrics are defined as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$$

$$\text{Recall (TPR)} = \frac{\text{TP}}{\text{TP+FN}}$$

$$\text{F-measure} = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision+Recall}}$$

$$\text{MCC} = \frac{\text{TP}\times\text{TN}-\text{FP}\times\text{FN}}{\sqrt{(\text{TP+FN})(\text{TP+FP})(\text{TN+FP})(\text{TN+FN})}}$$

Prior ML Models and Our Models. Previous works have focused on leveraging fundamental classification and regression techniques for disk failure prediction [4, 9, 51]. These methods include naive Bayes classifier (Bayes) [69], random forests (RF) [52], gradient boosted decision trees (GBDT) [29, 91] and long short-term memory networks (LSTM) [23, 38]. Bayes is a family of probabilistic classifiers based on applying Bayes' theorem. RF and GBDT are types of traditional machine learning (ML) ensemble methods, while LSTM is a class of deep neural networks (DNNs). Since previous works have not considered performance and location features for disk failure prediction, we implement and

tune Bayes, RF, GBDT, and LSTM models to use them as a proxy for prior learning based disk failure prediction models. In addition, we consider a convolutional neural network with long short-term memory (CNN-LSTM) based model [72]. We implement our models in Python, using TensorFlow 1.5.0 [1], Keras 2.1.5 [34], and Scikit-learn libraries [64] for model building.

Brief Model Background and Intuitions. Bayes [69] is a probabilistic machine learning model used for classification tasks. RF [52] and GBDT [29, 91] are both ensemble methods that are constructed by a multitude of individual trees (called base learners or weak learners) and consider the conclusions of all trees to make accurate predictions through averaging or max voting.

The difference between RF and GBDT is that RF generates trees in a parallel manner (bagging algorithm) [52], while GBDT grows tress sequentially (boosting algorithms) [29,91]. More specifically, the bagging algorithm randomly takes data samples with replacement from the original dataset to train every weak learner, which means that the training stage of generating multiple learners is parallel (i.e., each learner is built independently). Boosting algorithm, however, uses all data to train each learner and builds the new learner in a sequential manner, and it assigns more weight to the misclassified samples to pay more attention to improving their predictability them during the training phase.

On the other hand, LSTM [23,38] is capable of addressing the long-term back-propagation problem (iteratively adjusting the weights of network connections to reduce the value of the loss function). LSTM includes a memory cell which tends to preserve information for a relatively long time. Hence, LSTM is effective for sequential data modeling, and employing LSTM to predict disk failure has been explored previously [23]. To further improve the performance of LSTM in the disk failure prediction, we integrate CNN and LSTM as a unified CNN-LSTM model (a CNN at the front and an LSTM network at the rear), since CNN and LSTM are complementary in the modeling capabilities — CNN offers advantages in selecting better features, while LSTM is effective at learning sequential data [2]. The choice of combining CNN and LSTM is inspired by the analysis presented by Pascanu *et al.* [63] — suggesting that the performance of LSTM could be further improved by taking better features as the input, which could be provided by CNN through dimensionality reduction [68]. Therefore, we include the CNN-LSTM approach to explore its effectiveness in the field of disk failure prediction.

Model Training and Testing Methodology. We use 5-fold cross-validation [50], which is a validation technique to assess the predictive performance of machine learning models, judge how models perform to an unseen dataset (testing dataset) [70] and avoid the overfitting issue. More specifically, our dataset is randomly partitioned into five equal-sized sub-samples. We take one sub-sample as the testing dataset at a time and take the remaining four sub-samples as the training dataset. We fit a model on the training dataset, evaluate it on the testing dataset, and calculate the evaluation scores. After that, we retain the evaluation scores and discard the current model. The process is then repeated five times with different combinations of sub-samples, and we use the average of the five evaluation scores as the final result for each method.

Tuning Hyperparameters of Models. We search for the best values of hyperparameters for all models using the hold-out method [45], which splits our original training phase data further into the hyperparameter training dataset (80% of the original training phase data) and the validation dataset (20% of the original training phase data). The biggest difference between the hold-out method and k-fold cross-validation approach (k refers to the number of sub-samples) is that the training and validation process of the hold-out approach only needs to be run once, while k-fold cross-validation needs to be run k times. In the hyperparameter tuning phase, we conduct a grid search to build and evaluate models for each combination of hyperparameters, and the goal is to find the best combination with the highest performance. For example, for RF and GBDT, we run experiments with different numbers of trees (estimators), and we settle on using 2000 trees in the RF model, and 1000 trees in the GBDT model, since using more than 2000 and 1000 trees, respectively, does not have significant improvements in practice. Using a similar method, the additive Lidstone smoothing parameter (α) of Bayes [20] was set to 2.

For LSTM-based models, after conducting a grid search on the values of hyperparameters to find the best combinations, we build an LSTM model with four layers and 128 nodes. For CNN-LSTM, in the CNN sub-module, we employ 1 one-dimensional convolutional layer at the front followed by one max-pooling layer and one flatten layer (shown in Figure 6). The 1D convolutional layer contains 128 filters which interpret snapshots based on the input. The max-pooling layer is responsible for consolidating and abstracting the interpretation to get a two-dimensional matrix of features. The flatten layer transforms the matrix into a vector, which is fed into the next classifier. The LSTM module consists of two LSTM layers and one dense layer (fully connected layer). We empirically set the same learning rate of 0.001 for the LSTM and CNN-LSTM models, and we set the drop-out rate to 0.25.

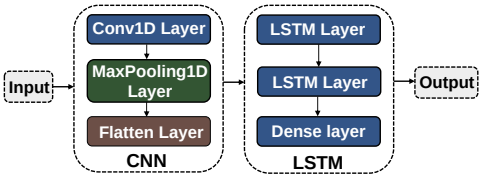

Figure 6: Structure of CNN-LSTM.

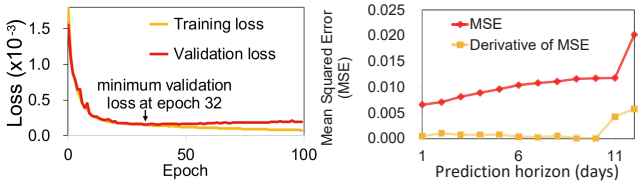

Figure 7: (a) The validation loss reaches its minimum value at 32 epochs for LSTM; thereafter it increases. (b) The mean squared error (MSE) and its derivative increases at a prediction horizon beyond 10 days.

Avoiding Overfitting of the Models. As far as LSTM and CNN-LSTM are concerned, one of the most important factors is the epoch [32], which indicates the number of iterations of processing the input dataset during the training process. A higher epoch value will reduce the error on training data; however, at a crucial tipping point, the network begins to over-fit the training data. Hence, finding the best value of the epoch is essential to avoid overfitting. Figure 7(a) shows the change in the value of the training and validation loss functions (the smaller, the better) as the epoch increases. Initially, the values of the two loss functions are decreasing with increasing epoch values; but after 32 epochs, the value of the validation loss function slowly increases (higher than the training loss), which indicates the over-fitting issue. Therefore, we choose 32 epochs for LSTM. Similarly, we choose 200 epochs for CNN-LSTM.

Feature Group Sets. We consider different input datasets to evaluate the effectiveness of different features: SMART attributes (S), performance metrics (P), and location markers (L). We construct six groups using different feature combinations: SPL, SL, SP, PL, S, and P. Table 7 shows the input features for these groups.

Prediction Horizon Selection. The first step in evaluating the ML model is to determine how long the prediction horizon should be. We choose 10 days as our prediction horizon, i.e., we aim to detect if a given disk will fail within the next 10 days, similar to previous studies [4, 9]. The 10-day horizon is long enough for IT operators to conduct early countermeasures. We also conduct a sensitivity study showing the change in the value of mean squared error (MSE) of different metrics

Table 7: Input features for six experimental groups. For performance metrics, the first column (red color) represents disk-level metrics, and the last two columns (yellow cells) represent server-level metrics.

	SMART	Performance			Location
SPL Group	28 attributes	12 metrics	18 metric categories	18 metrics	1 marker
SL Group	28 attributes	**NONE**			1 marker
SP Group	28 attributes	12 metrics	18 metric categories	18 metrics	**NONE**
PL Group	**NONE**	12 metrics	18 metric categories	18 metrics	1 marker
S Group	28 attributes	**NONE**			**NONE**
P Group	**NONE**	12 metrics	18 metric categories	18 metrics	**NONE**

for different lengths of prediction horizon, as shown in Figure 7(b) (using "ReadSuccessThroughput" as a representative example), where MSE indicates the average squared difference between the predicted values and the actual values [86]. We note that the derivative of MSE remains low for up to ten days, but it increases after ten days. This behavior can have slight variations across different features. Our prediction horizon is 10 days unless otherwise stated in our evaluation. We also evaluate the models' sensitivity with regard to prediction horizon (Section 5).

5 Results and Analysis

In this section, we present and analyze the results of various ML models, their sensitivity toward different feature groups, their limitations, robustness, and portability. Our discussion includes supporting evidence and reasons to explain observed trends, and implications of observed trends for data centers. First, we present the key prediction quality measures for all models and feature sets (Figure 8). We make several interesting observations as following:

1. We observe that the SPL feature group performs the best across all ML models, confirming our hypothesis that performance and location features are critical for improving the effectiveness of disk failure prediction, beyond traditional SMART attribute based approaches.

2. Adding location information improves the prediction quality across models, but the improvement is limited in absolute degree (e.g., less than 10% for CNN-LSTM in terms of MCC score). Interestingly, the effect of location information is pronounced only in the presence of performance features. The disk performance metrics are potentially correlated with disks' location information, Therefore, adding location markers may help ML mod-

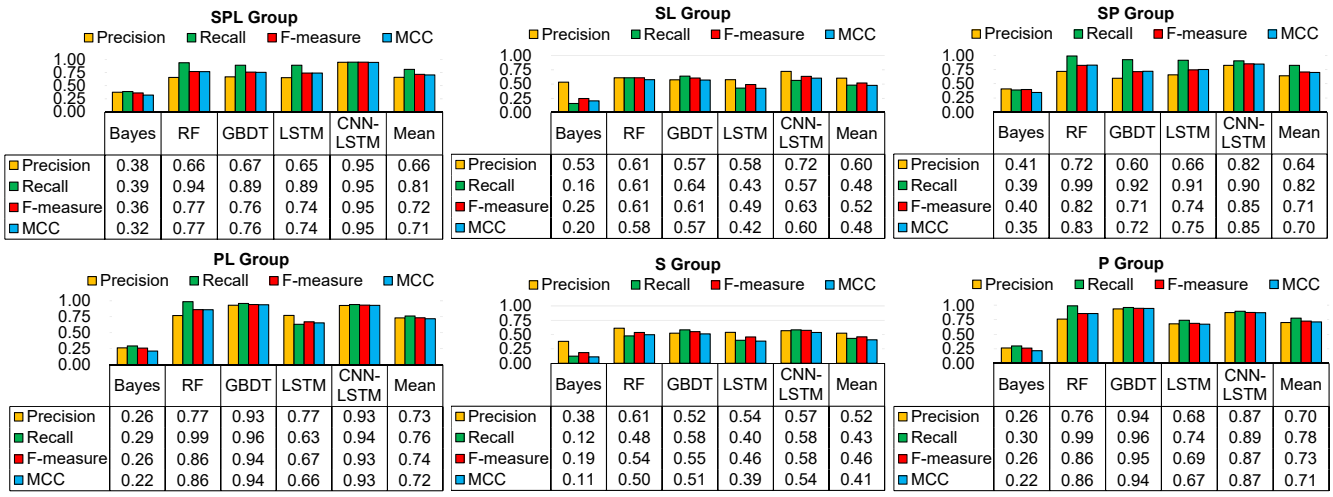

Figure 8: Model prediction quality with different groups of SMART (S), performance (P), and location (L) features.

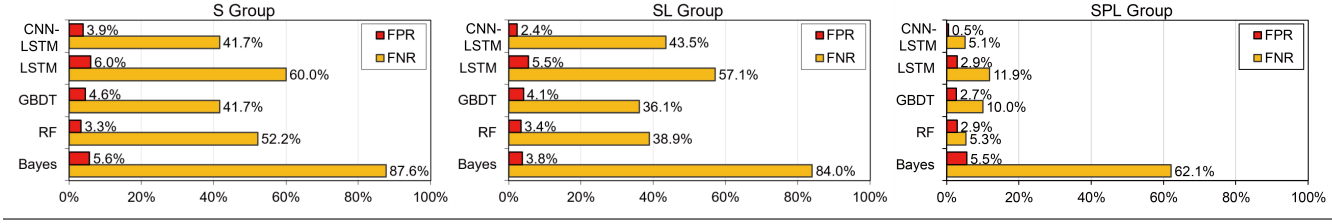

Figure 9: Model false positive rate ($FPR = FP/(FP + TN)$) and false negative rate ($FNR = FN/(TP + FN)$).

els amplify the hidden patterns in performance metrics.

3. While there is no single model winner across different feature groups, CNN-LSTM performs close to the best in all the situations, achieving an MCC score of 0.95 for the SPL group, compared to 0.77 MCC score for RF (next best method) for the SPL group. Further, we plot the false positive and false negative rates for different ML models for different feature groups (Figure 9). Figure 9 reveals interesting trends. First, SMART-attribute-based models have a very high false negative rate or FNR (failed disks predicted healthy) across all models. Adding performance and location features decreases the FNR significantly and hence, the prediction quality improves. It also decreases the false positive rate, but the scope for reduction is already limited.

Second, there is a trade-off between FPR and FNR in terms of cost (cost of disk failure vs. replacing healthy disks conservatively). Depending on the estimated costs of these factors, data center operators could choose between different models. For example, for the SPL group, GBDT provides lower FPR but higher FNR. Similarly, Figure 9 also shows that in the S group, such trade-offs exist between the RF and LSTM models.

4. Finally, we observe a trade-off between models with respect to the different availability of feature sets. Figure 8 shows that when a data center operator does not collect or have access to the performance features, traditional tree-based ML models (RF and GBDT) can perform roughly as well as complex neural network based models such as CNN-LSTM or LSTM. In fact, RF and GBDT models may even beat the LSTM model in absence of P and L features — this is similar to what a recent work has also shown which does not consider performance metrics [4].

Our work shows that adding performance and location features leads to a different and new outcome. Also, we note that the CNN-LSTM model takes much longer to train compared to simple tree-based models (up to four hours in our case for one training progress); therefore, in absence of performance and location features, RF and GBDT models can provide equally accurate predictions, and they might be preferred for building models based on the SMART data only due to the relatively lesser training time.

Next, we investigate when and how ML models fail to achieve high prediction accuracy over space and time.

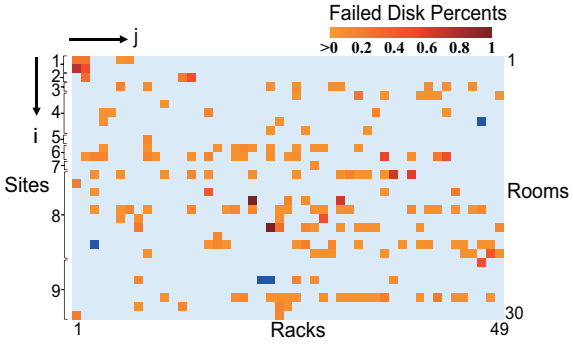

Figure 10: Mispredicted failures (blue) tend to occur in the locations where there is a low failure rate for all models. Each row stands for a room, and each column refers to a rack. i.e., the pixel of the j-th column and the i-th row represents the j-th rack of the i-th room. The color of each pixel indicates the failed disk percentage on the rack (pixel).

Where do ML models perform relatively poorly and why? Figure 10 shows that ML models are somewhat less effective at predicting with high accuracy and recall in areas where the concentration of failures is relatively lower. This is reasonable since ML models are not able to collect enough failed disk samples. ML models are by definition less effective for cases they have not been trained or situations they have not encountered before. This observation is important for data center operators as it emphasizes the need for adding location markers in disk failure prediction models.

When do ML models fail to predict and why? To understand the limitations of ML models better, we investigate the false positive (healthy disks predicted as failed) and false negative (failed disks predicted as healthy) predictions. Figure 11(a) shows the false positives categorized in 20-day windows for the CNN-LSTM model (other models produce similar trends). The number of false positives is very low initially as it predicts many disks as healthy though they eventually fail in that window — and, this is why the false negatives are high (Figure 11(b)). This can be explained by the lack of sufficient training data — the ML model does not have enough data and (conservatively) predicts that disks are healthy. This trend indeed reverses over time. Although the fraction of false positives appears to be very high toward the last window, we note that the actual number of false positives is quite low (Figure 9). This observation indicates the need for sufficiently long testing periods before concluding the prediction quality of ML models.

Is the prediction model portable across data center sites? Data centers operators often increase their number of sites over time, and it takes time to build models at

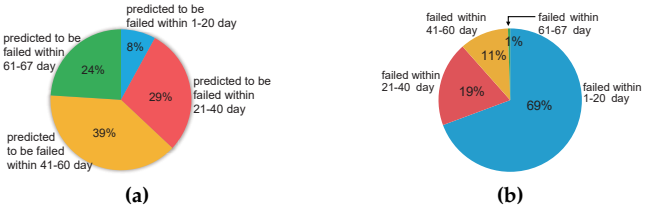

Figure 11: (a) Temporal distribution of CNN-LSTM model's false positives. (b) Temporal distribution of CNN-LSTM model's false negatives.

Table 8: Prediction quality on unseen Site A.

	Precision	Recall	F-measure	MCC
Bayes	0.35	0.37	0.36	0.31
RF	0.66	0.94	0.78	0.78
GBDT	0.65	0.89	0.75	0.74
LSTM	0.66	0.88	0.75	0.74
CNN-LSTM	0.93	0.94	0.94	0.93

new sites and in some cases, model training at new sites may not be possible due to strict business-sensitivity reasons. Therefore, we want to test if machine learning based disk failure models are unsuitable to a large degree for porting across data center sites? One can expect the operating conditions and workload characteristics to change across data center sites and hence, the disk failure prediction model may not work at all.

As expected, this is true if we simply try to train on one data center site and port it to another data center site (i.e., test on another unseen site) — the MCC score can drop significantly. However, we found that training on multiple data sites before testing on a new unseen data site provides reasonable accuracy. We tested on two unseen data center sites A and B, while training our model on the rest of the 62 sites (Table 8 shows results for site A; site B has similar results). Our results show that the prediction quality still remains reasonably high (e.g., >0.90 MCC score for a 10-day prediction horizon using CNN-LSTM model and SPL group features). We did not find a significant drop in prediction quality for any ML model; however, with some traditional ML models (RF and GBDT) the prediction quality does not remain high (more than 15% drop in some cases). Data center operators should be careful in porting ML-based prediction models as-is across sites without sufficiently training on multiple sites and should prefer CNN-LSTM models if portability is a requirement.

Is the prediction model effective at different prediction horizon (lead time)? To test this, we plotted MCC values for different ML models at different lead times (2-15 days). Figure 12 presents MCC scores of Bayes, RF, GBDT, LSTM, and CNN-LSTM for the SPL group, when the prediction horizon is 2 days, 5 days, 10 days and 15 days. As expected, the prediction quality indeed goes

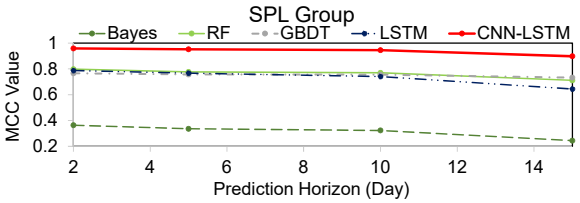

Figure 12: MCC scores of all ML models with SPL group features for different lengths of prediction horizon.

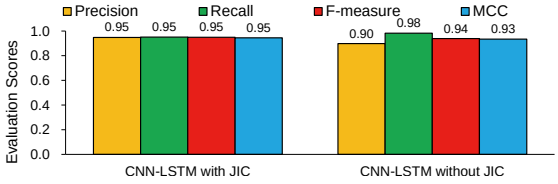

Figure 13: Prediction quality comparison among all features with and without J-Index classification (CNN-LSTM model on SPL group features).

down with increasing prediction horizon window (the MCC score for a 15-day window is 0.89), but the rate of decrease is not steep for any model — SPL group feature based ML models are effective even at sufficiently large prediction horizons.

Does J-Index classification for feature selection degrade the overall prediction accuracy compared to models trained with all features? Recall that we employed J-Index classification choosing the features (different performance and location metrics) for training our models. We compared the prediction quality for models using all the features (Figure 13). Our results show that manually selecting a subset of features using J-Index provides similar quality results, although it does affect the precision and recall trade-offs slightly. This notable observation suggests that data center operators can use J-Index to manage the storage overhead of storing attributes from thousands of disks without risking the prediction quality significantly.

6 Related Work

To the best of our knowledge, prior works do not consider all three types of data: SMART, performance, and location data for failure prediction. Instead, previous works rely only on SMART attributes [4, 19, 36, 41, 59, 79, 87]. We analyze large-scale field data collected from one of the biggest e-commerce sites, while most of the previous works propose prediction methods based on the publicly available Backblaze data [4, 5, 7, 9, 62, 80]. Also, the datasets analyzed were of limited in size, types of vendors, and were often closed-source [8, 9, 24, 28, 30, 31, 36, 37, 41, 51, 53, 58–60, 77, 83, 85, 88, 89, 92].

Much of previous work with disk failure prediction is limited to the detection of incipient failures [9, 41, 59, 67, 84, 85]. Although Lima *et al.* [23] proposed an approach to predict disk failures in long- and short-term, they are also limited to SMART attributes. Studies by Sandeep *et al.* [25, 26, 78] enable a qualitative understanding of factors that affect disk drive reliability. Yang *et al.* [90] and Gerry Cole [21] both achieve reliability predictions based on accelerated life tests. In addition, non-parametric statistical tests [58], Markov Models [24, 92], and Mahalanobis distance [85] have been proposed to predict disk failures. Hughes *et al.* [41] applied the multivariate rank-sum test and achieved a 60% failure detection rate (FDR).

In our study, we focus on HDDs, and some previous works have focused on solid-state drives (SSDs). Three typical studies of SSDs are based on data collected by Facebook [57], Google [77], and Alibaba Cloud [88]. Furthermore, Grupp *et al.* [33] examined the reliability of flash memory. Ouyang *et al.* [61] studied programmable SSD controllers at a web services company. A number of studies by Cai *et al.* [11–18] explored different patterns of Multi-Level Cell (MLC) flash chip failure. Ma *et al.* [53] found the accumulation of reallocated sectors would deteriorate disk reliability. Narayanan *et al.* [60] proposed machine learning based approaches to answer what, when and why of SSD failures.

Overall, few studies have separately employed ML [4, 36, 54, 79] and DNN techniques [4, 23] to predict disk failures. Our work explores and compares three classic ML methods with two DNNs using six feature groups to predict disk failures. This kind of extensive analysis helps us derive insights such as there is no need to employ complex DNNs when only SMART data are available. In fact, we are also the first to demonstrate the cross-site portability of different models.

7 Conclusion

We conducted a field study of HDDs based on a large-scale dataset collected from a leading e-commerce production data center, including SMART attributes, performance metrics, and location markers. We discover that performance metrics are good indicators of disk failures. We also found that location markers can improve the accuracy of disk failure prediction. Lastly, we trained machine learning models including neural network models to predict disk failures with 0.95 F-measure and 0.95 MCC for 10 days prediction horizon.

Acknowledgement

The authors are very thankful to the reviewers and our shepherd, Kimberly Keeton, for their constructive comments and suggestions. This work is supported in part by the National Science Foundation (NSF) grants CCF-1563728 and 1753840.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, volume 16, pages 265–283, 2016.

[2] Abdulaziz M Alayba, Vasile Palade, Matthew England, and Rahat Iqbal. A combined CNN and LSTM model for arabic sentiment analysis. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 179–191. Springer, 2018.

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

[4] Preethi Anantharaman, Mu Qiao, and Divyesh Jadav. Large scale predictive analytics for hard disk remaining useful life estimation. In *Proceedings of the 2018 IEEE International Congress on Big Data (BigData Congress)*, pages 251–254. IEEE, 2018.

[5] Nicolas Aussel, Samuel Jaulin, Guillaume Gandon, Yohan Petetin, Eriza Fazli, and Sophie Chabridon. Predictive models of hard drive failures based on operational data. In *Proceedings of the 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 619–625. IEEE, 2017.

[6] Viv Bewick, Liz Cheek, and Jonathan Ball. Statistics review 13: receiver operating characteristic curves. *Critical care*, 8(6):508, 2004.

[7] Shivam Bhardwaj, Akshay Saxena, and Achal Nayyar. Exploratory data analysis on hard drive failure statistics and prediction. *International Journal*, 6(6), 2018.

[8] Richard Black, Austin Donnelly, Dave Harper, Aaron Ogus, and Anthony Rowstron. Feeding the pelican: Using archival hard drives for cold storage racks. In *8th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage)*, 2016.

[9] 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)*, pages 39–48, 2016.

[10] Sabri Boughorbel, Fethi Jarray, and Mohammed El-Anbari. Optimal classifier for imbalanced data using matthews correlation coefficient metric. *PloS one*, 12(6):e0177678, 2017.

[11] Yu Cai, Saugata Ghose, Erich F Haratsch, Yixin Luo, and Onur Mutlu. Error characterization, mitigation, and recovery in flash-memory-based solid-state drives. *Proceedings of the IEEE*, 105(9):1666–1704, 2017.

[12] Yu Cai, Erich F Haratsch, Onur Mutlu, and Ken Mai. Error patterns in MLC NAND flash memory: Measurement, characterization, and analysis. In *Proceedings of the Conference on Design, Automation and Test in Europe (DATE)*, pages 521–526. EDA Consortium, 2012.

[13] Yu Cai, Erich F Haratsch, Onur Mutlu, and Ken Mai. Threshold voltage distribution in MLC NAND flash memory: Characterization, analysis, and modeling. In *Proceedings of the Conference on Design, Automation and Test in Europe (DATE)*, pages 1285–1290. IEEE, 2013.

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

[15] Yu Cai, Onur Mutlu, Erich F Haratsch, and Ken Mai. Program interference in MLC NAND flash memory: Characterization, modeling, and mitigation. In *Proceedings of the 31st International Conference on Computer Design (ICCD)*, pages 123–130. IEEE, 2013.

[16] Yu Cai, Gulay Yalcin, Onur Mutlu, Erich F Haratsch, Adrian Crista, Osman S Unsal, and Ken Mai. Error analysis and retention-aware error management for NAND flash memory. *Intel Technology Journal*, 17(1), 2013.

[17] Yu Cai, Gulay Yalcin, Onur Mutlu, Erich F Haratsch, Adrian Cristal, Osman S Unsal, and Ken Mai. Flash correct-and-refresh: Retention-aware error management for increased flash memory lifetime. In *Proceedings of the 30th International Conference on Computer Design (ICCD)*, pages 94–101. IEEE, 2012.

[18] Yu Cai, Gulay Yalcin, Onur Mutlu, Erich F Haratsch, Osman Unsal, Adrian Cristal, and Ken Mai. Neighbor-cell assisted error correction for MLC NAND flash memories. In *ACM SIGMETRICS Performance Evaluation Review (SIGMETRICS)*, volume 42, pages 491–504. ACM, 2014.

[19] Iago C Chaves, Manoel Rui P de Paula, Lucas GM Leite, Joao Paulo P Gomes, and Javam C Machado.

Hard disk drive failure prediction method based on a Bayesian network. In *Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2018.

[20] Stanley F Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 13(4):359–394, 1999.

[21] Gerry Cole. Estimating drive reliability in desktop computers and consumer electronics systems. *Seagate Technology Paper TP*, 338, 2000.

[22] Jesse Davis and Mark Goadrich. The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.

[23] 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 *Proceedings of the 2017 Brazilian Conference on Intelligent Systems (BRACIS)*, pages 222–227. IEEE, 2017.

[24] Ben Eckart, Xin Chen, Xubin He, and Stephen L Scott. Failure prediction models for proactive fault tolerance within storage systems. In *Proceedings of the 2008 IEEE International Symposium on Modeling, Analysis and Simulation of Computers and Telecommunication Systems (MASCOTS)*, pages 1–8. IEEE, 2008.

[25] Jon G Elerath and Sandeep Shah. Disk drive reliability case study: dependence upon head fly-height and quantity of heads. In *Proceedings of the 2003 Annual Reliability and Maintainability Symposium (RAMS)*, pages 608–612. IEEE, 2003.

[26] Jon G Elerath and Sandeep Shah. Server class disk drives: how reliable are they? In *Proceedings of the 2004 Annual Reliability and Maintainability Symposium (RAMS)*, pages 151–156. IEEE, 2004.

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

[28] Daniel Ford, François Labelle, Florentina Popovici, Murray Stokely, Van-Anh Truong, Luiz Barroso, Carrie Grimes, and Sean Quinlan. Availability in globally distributed storage systems. 2010.

[29] Jerome H Friedman. Stochastic gradient boosting. *Computational Statistics and Data Analysis*, 38(4):367–378, 2002.

[30] Peter Garraghan, Paul Townend, and Jie Xu. An empirical failure-analysis of a large-scale cloud computing environment. In *IEEE 15th International Symposium on High-Assurance Systems Engineering*, pages 113–120. IEEE, 2014.

[31] Moises Goldszmidt. Finding soon-to-fail disks in a haystack. In *HotStorage*, 2012.

[32] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6):602–610, 2005.

[33] Laura M Grupp, John D Davis, and Steven Swanson. The bleak future of NAND flash memory. In *Proceedings of the 10th USENIX conference on File and Storage Technologies (FAST)*, pages 2–2. USENIX Association, 2012.

[34] Antonio Gulli and Sujit Pal. *Deep Learning with Keras*. Packt Publishing Ltd, 2017.

[35] Chongomweru Halimu, Asem Kasem, and SH Newaz. Empirical comparison of area under roc curve (AUC) and matthews correlation coefficient (MCC) for evaluating machine learning algorithms on imbalanced datasets for binary classification. In *Proceedings of the 3rd International Conference on Machine Learning and Soft Computing*, pages 1–6. ACM, 2019.

[36] Greg Hamerly, Charles Elkan, et al. Bayesian approaches to failure prediction for disk drives. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, volume 1, pages 202–209, 2001.

[37] Mingzhe Hao, Gokul Soundararajan, Deepak Kenchammana-Hosekote, Andrew A Chien, and Haryadi S Gunawi. The tail at store: A revelation from millions of hours of disk and SSD deployments. In *Proceedings of the 14th USENIX Conference on File and Storage Technologies (FAST)*, pages 263–276, 2016.

[38] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[39] George Hripcsak and Adam S Rothschild. Agreement, the F-measure, and reliability in information retrieval. *Journal of the American Medical Informatics Association*, 12(3):296–298, 2005.

[40] Song Huang, Song Fu, Quan Zhang, and Weisong Shi. Characterizing disk failures with quantified disk degradation signatures: An early experience.

In *Proceedings of the 2015 IEEE International Symposium on Workload Characterization (IISWC)*, pages 150–159. IEEE, 2015.

[41] Gordon F Hughes, Joseph F Murray, Kenneth Kreutz-Delgado, and Charles Elkan. Improved disk-drive failure warnings. *IEEE transactions on reliability*, 51(3):350–357, 2002.

[42] Weihang Jiang, Chongfeng Hu, Yuanyuan Zhou, and Arkady Kanevsky. Are disks the dominant contributor for storage failures?: A comprehensive study of storage subsystem failure characteristics. *ACM Transactions on Storage (TOS)*, 4(3):7, 2008.

[43] Giuseppe Jurman, Samantha Riccadonna, and Cesare Furlanello. A comparison of MCC and cen error measures in multi-class prediction. *PloS one*, 7(8):e41882, 2012.

[44] Saurabh Kadekodi, KV Rashmi, and Gregory R Ganger. Cluster storage systems gotta have heart: improving storage efficiency by exploiting disk-reliability heterogeneity. In *Proceedings of the 17th USENIX Conference on File and Storage Technologies (FAST)*, pages 345–358, 2019.

[45] Ji-Hyun Kim. Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Computational statistics and data analysis*, 53(11):3735–3745, 2009.

[46] Andy Klein. What SMART stats tell us about hard drives. https://www.backblaze.com/blog/what-smart-stats-indicate-hard-drive-failures/, October 2016.

[47] Andy Klein. Backblaze hard drive stats for 2017. https://www.backblaze.com/blog/hard-drive-stats-for-2017/, February 2018.

[48] Andy Klein. Backblaze hard drive stats for 2018. https://www.backblaze.com/blog/hard-drive-stats-for-2018/, January 2019.

[49] Andy Klein. Backblaze hard drive stats Q3 2019. https://www.backblaze.com/blog/backblaze-hard-drive-stats-q3-2019/, November 2019.

[50] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artificial Intelligence (IJCAI)*, volume 14, pages 1137–1145, 1995.

[51] 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 *Proceedings of the 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 383–394. IEEE, 2014.

[52] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomForest. *R news*, 2(3):18–22, 2002.

[53] 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. In *Proceedings of the 13th USENIX Conference on File and Storage Technologies (FAST)*, volume 11, page 17, 2015.

[54] Farzaneh Mahdisoltani, Ioan Stefanovici, and Bianca Schroeder. Proactive error prediction to improve storage system reliability. In *Proceedings of the 2017 USENIX Annual Technical Conference (ATC). Santa Clara, CA*, pages 391–402, 2017.

[55] Ioannis Manousakis, Sriram Sankar, Gregg McKnight, Thu D Nguyen, and Ricardo Bianchini. Environmental conditions and disk reliability in free-cooled datacenters. In *Proceedings of the 12th USENIX Conference on File and Storage Technologies (FAST)*, pages 53–65, 2016.

[56] Brian S Merrow. Vibration isolation within disk drive testing systems, November 6 2012. US Patent 8,305,751.

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

[58] Joseph F Murray, Gordon F Hughes, and Kenneth Kreutz-Delgado. Hard drive failure prediction using non-parametric statistical methods. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, 2003.

[59] 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(May):783–816, 2005.

[60] 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)*, page 7. ACM, 2016.

[61] Jian Ouyang, Shiding Lin, Song Jiang, Zhenyu Hou, Yong Wang, and Yuanzheng Wang. SDF: software-defined flash for web-scale internet storage systems. In *ACM SIGARCH Computer Architecture News (ASPLOS)*, volume 42, pages 471–484. ACM, 2014.

[62] Jehan-François Pâris, SJ Thomas Schwarz, SJ Ahmed Amer, and Darrell DE Long. Protecting RAID arrays against unexpectedly high disk failure rates. In *Proceedings of the IEEE 20th Pacific Rim International Symposium on Dependable Computing (PRDC)*, pages 68–75. IEEE, 2014.

[63] Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. How to construct deep recurrent neural networks. *arXiv preprint arXiv:1312.6026*, 2013.

[64] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[65] Eduardo Pinheiro, Wolf-Dietrich Weber, and Luiz André Barroso. Failure trends in a large disk drive population. In *Proceedings of the 5th USENIX Conference on File and Storage Technologies (FAST)*, volume 7, pages 17–23, 2007.

[66] David Martin Powers. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. Bioinfo Publications. 2011.

[67] 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 non-parametric statistics. *IEEE Transactions on Industrial Informatics*, 13(2):542–550, 2017.

[68] Oren Rippel, Jasper Snoek, and Ryan P Adams. Spectral representations for convolutional neural networks. In *Advances in neural information processing systems*, pages 2449–2457, 2015.

[69] Irina Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

[70] Juan D Rodriguez, Aritz Perez, and Jose A Lozano. Sensitivity analysis of K-fold cross validation in prediction error estimation. *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, 32(3):569–575, 2010.

[71] Kunal Roy, Supratik Kar, and Rudra Narayan Das. *Understanding the basics of QSAR for applications in pharmaceutical sciences and risk assessment*. Academic press, 2015.

[72] Tara N Sainath, Oriol Vinyals, Andrew Senior, and Haşim Sak. Convolutional, long short-term memory, fully connected deep neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4580–4584. IEEE, 2015.

[73] Sriram Sankar, Mark Shaw, and Kushagra Vaid. Impact of temperature on hard disk drive reliability in large datacenters. In *Proceedings of IEEE/IFIP the 41st International Conference on Dependable Systems and Networks (DSN)*, pages 530–537. IEEE, 2011.

[74] Enrique F Schisterman, Neil J Perkins, Aiyi Liu, and Howard Bondell. Optimal cut-point and its corresponding Youden Index to discriminate individuals using pooled blood samples. *Epidemiology*, pages 73–81, 2005.

[75] Bianca Schroeder, Sotirios Damouras, and Phillipa Gill. Understanding latent sector errors and how to protect against them. *ACM Transactions on storage (TOS)*, 6(3):9, 2010.

[76] Bianca Schroeder and Garth A Gibson. Disk failures in the real world: What does an MTTF of 1, 000, 000 hours mean to you? In *Proceedings of the 5th USENIX Conference on File and Storage Technologies (FAST)*, volume 7, pages 1–16, 2007.

[77] Bianca Schroeder, Raghav Lagisetty, and Arif Merchant. Flash reliability in production: The expected and the unexpected. In *Proceedings of the 14th USENIX Conference on File and Storage Technologies (FAST)*, pages 67–80, 2016.

[78] Sandeep Shah and Jon G Elerath. Reliability analysis of disk drive failure mechanisms. In *Proceedings of the 2005 Annual Reliability and Maintainability Symposium (RAMS)*, pages 226–231. IEEE, 2005.

[79] 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), 2018.

[80] Chuan-Jun Su and Shi-Feng Huang. Real-time big data analytics for hard disk drive predictive maintenance. *Computers and Electrical Engineering*, 71:93–101, 2018.

[81] Yoshimasa Tsuruoka and Jun'ichi Tsujii. Boosting precision and recall of dictionary-based protein name recognition. In *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine-Volume 13*, pages 41–48. Association for Computational Linguistics, 2003.

[82] Kashi Venkatesh Vishwanath and Nachiappan Nagappan. Characterizing cloud computing hardware reliability. In *Proceedings of the 1st ACM symposium on Cloud computing (SoCC)*, pages 193–204. ACM, 2010.

[83] Guosai Wang, Lifei Zhang, and Wei Xu. What can we learn from four years of data center hardware failures? In *Proceedings of the 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 25–36. IEEE, 2017.

[84] Yu Wang, Eden WM Ma, Tommy WS 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, 2014.

[85] 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.

[86] Zhou Wang and Alan C Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE signal processing magazine*, 26(1):98–117, 2009.

[87] 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*, page 35. ACM, 2018.

[88] 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 *Proceedings of 2019 USENIX Annual Technical Conference (ATC)*, pages 961–976, 2019.

[89] 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 *Proceedings of 2018 USENIX Annual Technical Conference (ATC)*, pages 481–494, 2018.

[90] Jimmy Yang and Feng-Bin Sun. A comprehensive review of hard-disk drive reliability. In *Proceedings of Annual Reliability and Maintainability Symposium (RAMS)*, pages 403–409. IEEE, 1999.

[91] Jerry Ye, Jyh-Herng Chow, Jiang Chen, and Zhaohui Zheng. Stochastic gradient boosted distributed decision trees. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 2061–2064. ACM, 2009.

[92] Ying Zhao, Xiang Liu, Siqing Gan, and Weimin Zheng. Predicting disk failures with HMM-and HSMM-based approaches. In *Proceedings of the 2010 Industrial Conference on Data Mining (ICDM)*, pages 390–404, 2010.