SSD Failures in the Field: Symptoms, Causes, and Prediction Models

Jacob Alter
College of William & Mary
Williamsburg, VA, USA
jralter@email.wm.edu

Ji Xue
College of William & Mary
Williamsburg, VA, USA
xuejimic@cs.wm.edu

Alma Dimnaku
NetApp, Inc.
Waltham, MA, USA
Alma.Dimnaku@netapp.com

Evgenia Smirni
College of William & Mary
Williamsburg, VA, USA
esmirni@cs.wm.edu

ABSTRACT

In recent years, solid state drives (SSDs) have become a staple of high-performance data centers for their speed and energy efficiency. In this work, we study the failure characteristics of 30,000 drives from a Google data center spanning six years. We characterize the workload conditions that lead to failures and illustrate that their root causes differ from common expectation but remain difficult to discern. Particularly, we study failure incidents that result in manual intervention from the repair process. We observe high levels of infant mortality and characterize the differences between infant and non-infant failures. We develop several machine learning failure prediction models that are shown to be surprisingly accurate, achieving high recall and low false positive rates. These models are used beyond simple prediction as they aid us to untangle the complex interaction of workload characteristics that lead to failures and identify failure root causes from monitored symptoms.

CCS CONCEPTS

•Information systems → Data centers; •Computing methodologies → Supervised learning; Supervised learning by classification; *Classification and regression trees;* **•Computer systems organization → Dependable and fault-tolerant systems and networks; Reliability; •Hardware → Fault tolerance; Failure prediction;**

ACM Reference format:
Jacob Alter, Ji Xue, Alma Dimnaku, and Evgenia Smirni. 2019. SSD Failures in the Field: Symptoms, Causes, and Prediction Models . In *Proceedings of The International Conference for High Performance Computing, Networking, Storage, and Analysis, Denver, CO, USA, November 17–22, 2019 (SC '19),* 13 pages.
DOI: 10.1145/3295500.3356172

SC '19, Denver, CO, USA

978-1-4503-6229-0/19/11…$15.00
DOI: 10.1145/3295500.3356172

1 INTRODUCTION

The longevity of solid state drives (SSDs) and their reliability is of much importance in recent years as increasing amounts of data in modern data centers reside on SSDs. Unfortunately, there are relatively few studies on the reliability of SSD devices, especially compared to the extensive studies of hard disk drives (HDDs) occurring over the past several decades. HDD reliability research is not generalizable to SSDs as the physical mechanics of HDDs are distinct from SSDs and, correspondingly, their failure symptoms and causes are fundamentally different. For the few existing SSD reliability analyses, their scope has focused more on specific errors in controlled, laboratory environments using simulated workloads [4, 10, 15, 23, 32]. Non-simulated studies of SSD reliability analysis, centered around production systems, typically focus on the type of errors (in particular, raw bit error rates and uncorrectable bit error rates), their relationship with the workload, drive age, and drive wear out. However these results also extend to field characteristics of block failures, chip failures, and rates of repairs and replacements [17, 18, 25].

In this paper we look into the process of SSD retirements by examining those drive failures that necessitate manual intervention and repairs. We study this through analysis of a selection of daily performance logs for three multi-level cell (MLC) models collected at a Google production data center over the course of six years. We concern ourselves with the conditions of drive activity that precede these total failures. Though we are unaware of the data center's exact workflow for drive repairs and replacements (e.g., whether they are done manually or automatically, or the replacement policies in place), we are able to discover key correlations and patterns of failure, as well as generate useful forecasts of future failures. Being able to predict an upcoming retirement could allow early action: for example, early replacement before failure happens, migration of data and VMs to other resources, or even allocation of VMs to disks that are not prone to failure [31].

In this paper, we study the various error types accounted by the logs to determine their roles in triggering, or otherwise portending, future drive failures. It is interesting to note that although we have ample data, statistical methods are not able to achieve highly accurate predictions: we find no evidence that the repair process is triggered by any deterministic decision rule. Since the complexity of the data does not allow for a typical treatment of prediction based on straightforward statistical analysis, we resort to machine

learning predictors to help us detect which quantitative measures provide strong indications of upcoming failures. We show that machine learning models that are trained from SSD monitoring logs achieve failure prediction that is both remarkably accurate and timely. Beyond prediction, the models are interpreted to provide valuable insights on which errors and workload characteristics are most indicative of future catastrophic failures.

The models are able to anticipate failure events with reasonable accuracy up to several days in advance, despite some inherent methodological challenges. Although the frequency of failures is significant (14% of the drives experience failures during their lifetime), the data set is highly imbalanced. This is a common problem in classification, and makes achieving simultaneously high true positive rates and low false positive ones very difficult. We train a set of six machine learning predictors and illustrate that it is possible to achieve robust predictions. Crucially, we focus on the interpretability of the machine learning models and derive insights that can be used to drive proactive SSD management policies. Our findings are summarized as follows:

- Although a significant portion of drives (up to 14%) are swapped during their lifetime, a very small percentage of swapped drives that go into repair mode re-enter the workflow within a month, this percentage is as low as 5% and as high as 9.4%, depending on the drive type. Up to 28% of drives are repaired within a year and about 50% of those that are swapped are never put back into production.
- A significant proportion of failed drives (roughly 10%) remain in the system in a failed state for a period on the order of months.
- There is no single metric that triggers a drive failure after it reaches a certain threshold.
- Several different machine learning predictors are quite successful for failure prediction. Random forests are found to be the most successful of all.
- We identify the drive age as the most important feature for swap prediction (and also for non-transparent error prediction).
- We are unable to demonstrate a correspondence between P/E cycles and failure, suggesting that write behavior is not as highly indicative of failure as previously thought.
- Different predictors need to be trained for drives of different age groups and the which features are useful for failure prediction depends heavily on the age of the drive.

The above insights can be used to anticipate failures and take appropriate actions such as proactive SSD management, spare drive provisioning, and perhaps even workload allocation.

This paper is organized as follows. We first characterize the data and summarize our findings in Section 2 and 3, respectively. Section 4 connects statistics between failure statistics to workload statistics to identify symptoms and causes of drive failures. In Section 5, we propose several machine learning predictors of SSD failure and conduct detailed post-prediction analysis. Section 6 presents related work, followed by the summary and conclusions in Section 7.

2 SSD TRACE DATA

The data consist of daily performance logs for three MLC SSD models collected at a Google data center over a period of six years. Each of the three models are manufactured by the same vendor and have a capacity 480GB and a lithography on the order of 50nm. All three models utilize custom firmware and drivers, meaning that error reporting is done in a proprietary format rather than through standard SMART features [1]. We refer to the three models as MLC-A, MLC-B, and MLC-D in accordance with the naming in [17, 25]. We have data on over 10,000 unique drives for each drive model, totaling over 40,000,000 daily drive reports overall.

The logs used in this paper report daily summaries of drive activity. Drives are uniquely identified by their drive ID, which is a hashed value of their serial number. For each day of operation, the following metrics are reported:

- The timestamp of the report, given in microseconds since the beginning of the drive's lifetime
- The number of read, write, and erase operations performed by the drive over the course of the day
- The cumulative number of program–erase (P/E) cycles seen by the drive over its lifetime. A program–erase cycle is the process by which a memory cell is written to and subsequently erased. The cumulative amount of these cycles is a measure of device wear.
- Two status flags indicating whether the drive has died and whether the drive is operating in read-only mode.
- The number of *bad blocks* in the drive. A block is marked bad either when it is non-operational upon purchase (denoted a *factory bad block*) or when a non-transparent error occurs in the block and it is subsequently removed from use. Cumulative counts of both of these bad block types are provided in the log.
- The counts of different errors that have occurred over the course of the day, specific counts are provided for the following error types:
  - *correctable error*: the total number of bits that were found corrupted and corrected using drive-internal error correction codes (ECC) during read operations during that day,
  - *erase error*: number of erase operations that failed,
  - *final read error*: the number of read operations that fail, even after (drive-initiated) retries,
  - *final write error*: the number of write operations that fail, even after (drive-initiated) retries,
  - *meta error*: number of errors encountered while reading drive-internal metadata,
  - *read error*: the number of read operations that experienced an error, but succeeded on retry (initiated drive-internally),
  - *response error*: number of bad responses from the drive,
  - *timeout error*: number of operations that timed out after some wait period,
  - *uncorrectable error*: number of uncorrectable ECC errors encountered during read operations during that day, and
  - *write error*: the number of write operations that experienced an error, but succeeded on retry (initiated drive-internally).

For a given drive, the error log may have observations spanning over a period of several days up to several years. This is demonstrated in the "Max Age" CDF in Figure 1, which shows the distribution over "oldest" observations we have for each drive. This measure indicates the length of the observational horizons we possess.

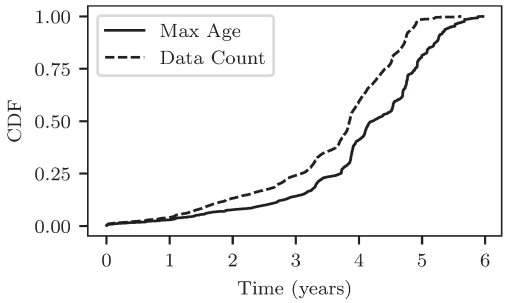

Figure 1: CDFs of maximum observed drive age (solid) and number of observed drive days within the error log (dashed) for each drive.

Error type	MLC-A	MLC-B	MLC-D
correctable error	0.828895	0.776308	0.767593
final read error	0.001077	0.001805	0.001552
final write error	0.000026	0.000027	0.000034
meta error	0.000014	0.000016	0.000028
read error	0.000090	0.000103	0.000133
response error	0.000001	0.000004	0.000002
timeout error	0.000009	0.000010	0.000014
uncorrectable error	0.002176	0.002349	0.002583
write error	0.000117	0.001309	0.000162

Table 1: Proportion of drive days that exhibit each error type

We observe that, for over 50% of drives, we have data extending over a period of 4 to 6 years. However, some of these days of drive activity are not recorded in the log. Accordingly, we may ask what magnitude of data we have access to for a given drive. The accompanying "Data Count" CDF shows exactly this: the number of drive days that are recorded in the error log for each drive. "Data Count" is a measure of the volume of the log entries. Measuring them as a function of time is reasonable as there is one log entry per day per drive. The data shown in Figure 1 clearly shows that there are ample data available, and therefore amenable to the analysis presented in this paper.

The errors collected in the error log can be separated into two types: *transparent* and *non-transparent* errors. Transparent errors (i.e., correctable, read, write, and erase errors) may be hidden from the user while non-transparent errors (i.e., final read, final write, meta, response, timeout, and uncorrectable errors) may not. Incidence statistics for each of these error types are listed in Table 1. Note that meta, response, and timeout errors are very rare. Uncorrectable errors and final read errors are more common at least by one order of magnitude comparing to other errors. For a detailed analysis of this data set focusing on raw bit error rates, uncorrectable bit error rates, and their relationships with workload, age, and drive wear out, we direct the interested reader to [25].

SSDs can only experience a finite number of write operations before their cells begin to lose their ability to hold a charge. Hence, manufacturers set a limit to the number of P/E cycles a given drive model can handle. For our drive models, this limit is 3000 cycles. Due to these limits, it is believed that errors are caused in part by wear of write operations on the drive, which one can measure using either a cumulative P/E cycle count or a cumulative count of write operations. Using either measure is equivalent since they are very highly correlated.

Table 2 illustrates the Spearman correlation matrix across pairs of all measures, aiming to determine whether there are any strong co-incidence relationships between different error types. Spearman correlations are used as a non-parametric measure of correlation, with values ranging between −1 and +1. The Spearman correlation differs from the more common Pearson correlation in that it is able to detect all sorts of monotonic relationships, not just linear ones [5]. In Table 2, we are especially interested in *non-transparent* errors because they are those that are most indicative of aberrant behavior. Bolded values are those with magnitude greater than or equal to 0.30, indicating a non-negligible relationship between the pair.

It is interesting to observe that there is little-to-no correlation of P/E cycle count with any of the other errors, except for some moderate correlation with erase errors. which contradicts common expectations. One reason for this is the aforementioned argument regarding device wear. Another is that drives which experience more activity will simply have more opportunities for errors to occur. We are unable to detect any substantial effects even due to this naïve relationship. Note that the correlation value of P/E cycle count and uncorrectable error count (which reflects bad sectors and eventual drive swap) is mostly insignificant. The age of a drive gives a similar metric for drive wear, which correlates highly with the P/E cycle count. The drive age also has very small correlation with cumulative error counts, with the exception of uncorrectable/final read errors. Bad blocks, another likely culprit for drive swaps, shows some mild correlation with erase errors, final read errors, and uncorrectable errors. The high value of the correlation coefficient of 0.97 between uncorrectable errors and final read errors is not useful as the two errors represent essentially the same event: if a read fails finally, then it is uncorrectable. Yet, we see some moderate correlation values between certain pairs of measures that eventually show to be of use for swap prediction within the context of machine learning-based predictors, see Section 5.

Observation #1: There is no clear relationship between non-transparent error types and uncorrectable error counts that presumably result in bad sectors and eventual drive failures. Program–erase (P/E) cycle counts, an indicator of drive wear, show very low correlations with most uncorrectable errors and mild correlation with erase errors (transparent errors). Drive age shows a similar pattern of correlation.

Observation #2: Correlations among all pairs of transparent and non-transparent errors show that some pairs may be mildly correlated and can be useful in prediction. Yet, there is no strong indication as to which measures are most useful for prediction.

3 DRIVE SWAP AND REPAIR

In addition to daily performance metrics, special "swap" events are also reported in the data. These events indicate the time at which

	erase	final read	final write	meta	read	response	timeout	uncorrect.	write	P/E cycle	bad block
erase	1.00										
final read	0.21	1.00									
final write	0.24	0.12	1.00								
meta	0.17	0.19	**0.35**	1.00							
read	0.22	0.20	0.30	**0.40**	1.00						
response	0.02	0.06	0.24	0.02	0.03	1.00					
timeout	0.01	0.12	**0.44**	0.02	0.03	**0.53**	1.00				
uncorrectable	0.20	**0.97**	0.06	0.16	0.15	0.03	0.03	1.00			
write	**0.32**	0.28	0.13	0.14	0.25	0.02	0.02	0.28	1.00		
P/E cycle	**0.32**	0.18	−0.05	−0.02	0.03	0.03	0.00	0.19	0.23	1.00	
bad block count	**0.38**	**0.37**	0.19	0.19	0.18	0.01	0.01	**0.37**	**0.34**	0.16	1.00
drive age	0.20	**0.36**	0.06	0.05	0.06	0.04	0.05	**0.36**	0.14	**0.73**	0.18

Table 2: Matrix of Spearman correlations among cumulative error counts and cumulative P/E cycle count. Bolded text indicates a large correlation value.

Model	#Failures	%Failed
MLC-A	734	6.95
MLC-B	1565	14.3
MLC-D	1580	12.5
All	3879	11.29

Table 3: High-level failure incidence statistics. This includes, for each model, the number of failures observed and the proportion of drives that are observed to fail at least once.

failed drives are extracted to be repaired. Swaps denote visits to the repairs process — and not simply a swapping out for storing spare parts, or moving a healthy SSD to a storage cabinet. All swaps follow drive failures, and accordingly, each swap documented in the log corresponds to a single, catastrophic failure. Incidence statistics for these swaps/failures are provided in Table 3.

Number of Failures	% of drives	% of failed drives
0	88.71	—
1	10.10	89.60
2	1.038	9.208
3	0.133	1.180
4	0.001	0.001

Table 4: Distribution of lifetime failure counts. The distribution is expressed with respect to the entire population of drives and with respect to those drives which fail at least once ("failed drives").

Table 3 shows that failures occur to a significant proportion of drives and are a relatively common occurrence in this data center: a whole 14.3% of MLC-B drives have failed at least once, followed by 12.5% of MLC-D drives and 6.95% of MLC-A ones. This high frequency of failures/swaps poses a large pressure in terms of maintenance costs, since each swap requires manual intervention. Table 4 provides more insights by providing statistics on the frequency of failures for the *same* drive. Unexpectedly, we find that some drives have failed as many as four times over the course of their lifetime. Nonetheless 89.6% of drives that have been swapped, are swapped only once.

To better characterize the conditions of failure that lead to these repairs, we must pinpoint the failures in the timeline. A natural way to proceed is to define failure events with respect to swap events: a failure occurs on a drive's last day of *operational activity* prior to a swap. This is a natural point of failure since, after this point in the timeline, the drive has ceased normal function and is soon be sent to the repairs process.

We now discuss what we consider to be "operational activity." It is often the case (roughly 80% of the time) that swaps are preceded by at least one day for which *no performance summaries* are documented in the log. This indicates that the drive was non-operational during this period, having suffered a complete failure. Prior to this period, we also find substantially higher rates of inactivity relatively to normal drive operation. In this case, inactivity refers to an absence of read or write operations provisioned to the drive. A period of inactivity like this is experienced prior to 36% of swaps. The length of these inactive periods is less than one week in a large majority of cases. The existence of such inactivity is an indication that data center maintainers no longer provision workloads onto the drive: this amounts to a "soft" removal from production before the drive is physically swapped. Accordingly, we define a failure as happening directly prior to this period of inactivity, if such a period exists.

To summarize, drive repairs undergo the following sequence of events, represented in Figure 2: 1) At some point, the drive undergoes a failure, directly after which the drive may cease read/write activity, cease to report performance metrics, or both, in succession. 2) Data center maintenance takes notice of the failure and swap the faulty drive with an alternate. Such swaps are notated as special

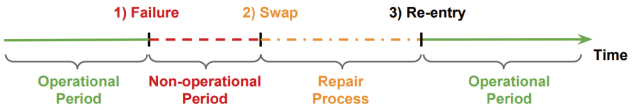

Figure 2: Overview of failure timeline

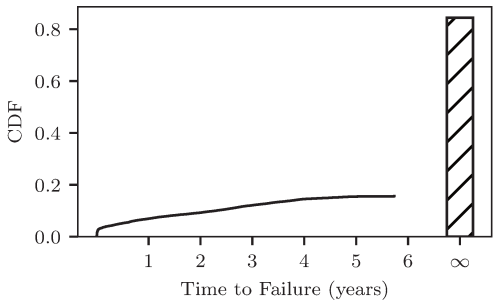

Figure 3: CDF of the length of the drive's operational period. The bar indicates what proportion of operational periods are not observed to end.

events in the data. 3) After a visit to the repairs process, a drive may or may not be returned to the field to resume normal operation.

We will now characterize each of these three stages, in turn. We start by examining the operational periods observed in the swap log. Figure 3 presents the CDF of the length of operational periods (alternately denoted "time to failure"). The CDF includes both operational periods starting from the beginning of the drive's lifetime and operational periods following a post-swap re-entry. It is interesting to note that more than 80% of the operational periods are not observed to end in failure during the 6 year sampling period; this probability mass is indicated by the bar centered at infinity. The figure indicates that there is substantial variability in the drive operational time, with the majority of operating times being long. Yet, there is a non-negligible portion of operating times that are interrupted by failures.

We now turn to the non-operational periods. In Figure 4, the CDF of the length of the pre-swap non-operational period is shown, i.e., the elapsed time between the drive failure and when it is swapped out of production. One can see that roughly 20% of failed drives are removed within a day and roughly 80% of failed drives are swapped out of the system within 7 days. However, this distribution has a very long tail (note the logarithmic scale on the x-axis). A non-negligible proportion of failed drives (roughly 8%) remain in a failed state for upwards of 100 days before they are removed from production. Since these faulty drives can remain in limbo for upwards of a year, the data suggest that these drives may simply have been forgotten in the system.

Similarly, we turn to Figure 5 for the CDF of the length of the repairs process, a.k.a. the "time to repair." We find that half of drives are never observed to re-enter the field (i.e., the time to repair is infinite – their share of probability mass is again indicated by the bar). Furthermore, among drives that *are* returned to the field, a

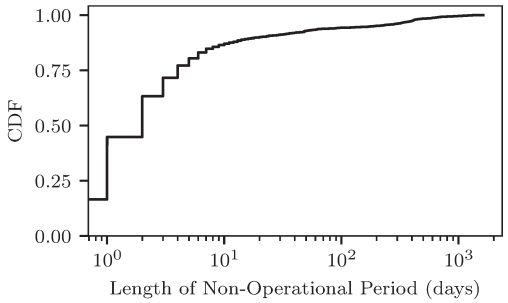

Figure 4: CDF of length of non-operational period preceding a swap. This is the number of days between the swap-inducing failure and the physical swap itself.

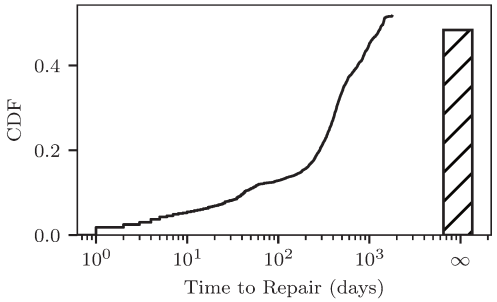

Figure 5: CDF of time to repair, ranging from 1 day to 4.85 years. The proportion of repairs that are not observed to terminate is depicted with the bar graph.

majority of them remain in the repairs process for upwards of a year with a maximum repair time of 4.85 years.

Restating the results in Figure 5, Table 5 illustrates the percentage of swapped drives that are repaired and re-enter the system after a period of n days in repair, the percentage of the successfully repaired drives as a function of all drives is also shown within the parentheses. These metrics demonstrate that repairs are very slow and that only half of the swapped drives are eventually returned to production by the end of the 6-year trace period.

Observation #3: Failed drives are often swapped out of production within a week, though a small portion may remain in the system even longer than a year.

Observation #4: While a significant percentage of drives (up to 14.3% for MLC-B, slightly smaller percentages for the other MLC types) are swapped during their lifetime, only half of failed drives are seen to successfully complete the repairs process and re-enter the field.

Observation #5: Of those repairs that do complete, only a small percentage of them finish within 10 days. About half of drives that are swapped out are not successfully repaired.

Model	10 days	30 days	100 days	1 year	2 years	3 years	∞
MLC-A	3.4 (0.23)	5.0 (0.34)	6.1 (0.43)	17.4 (2.61)	37.6 (2.61)	43.6 (3.03)	53.4 (3.71)
MLC-B	6.8 (0.98)	9.4 (1.34)	12.7 (1.81)	25.3 (3.62)	36.1 (5.16)	42.7 (6.11)	43.9 (6.28)
MLC-D	4.9 (0.61)	8.1 (1.01)	15.8 (1.97)	28.1 (3.51)	43.5 (5.44)	50.2 (6.28)	57.6 (7.20)

Table 5: Percentage of swapped drives that re-enter the workflow within n days. The percentage of repaired drives as a function of all drives is also reported within the parentheses.

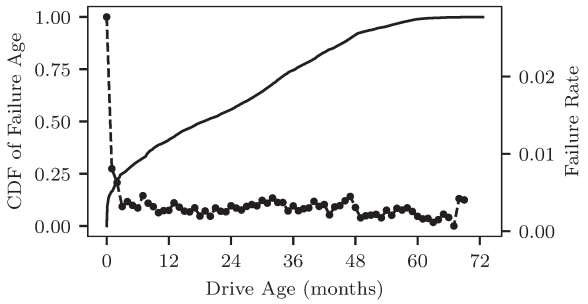

Figure 6: The CDF of the age of failed drives (solid line) and the proportion of functioning drives that fail at a given age level, in months (dashed line).

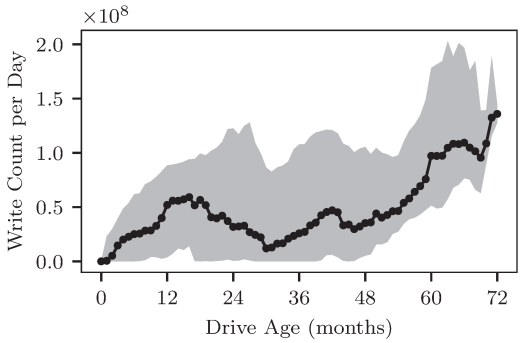

Figure 7: Quartiles of the daily write intensity per month of drive age. The line plot shows the median write intensity for each month. The 1st and 3rd quartiles are shown as the boundaries of the shaded area.

4 SYMPTOMS AND CAUSES OF SSD FAILURES

4.1 Age and Device Wear

In this section we make an effort to connect the statistics from the two logs, aiming to identify causes of drive failures. Recall that Table 4 shows that, among the drives represented in the error log, 11.29% of them are swapped at least once. A natural question is when do these swaps (and the preceding failures) occur in the drive's lifetime: what is the role of age in drive failure? Figure 6 reports the CDF of the failure age (solid line) as a function of the drive age. The figure shows that there are many more drive failures in the first 90 days of drive operation than at any other point in the drive's lifetime. In fact, 15% of observed failures occur on drives less than 30 days old and 25% occur on drives less than 90 days old. This seems to indicate that these drives have an infancy period during which drive mortality rate is particularly high. This performance pattern has been noticed previously in similar studies of SSDs in the wild [18].

The slope of the CDF in Figure 6 gives us an estimate of the rate at which swaps occur at a given drive age. However, this estimate is skewed since not all drive ages are equally represented in the data. For example, the rate of failures seems to slow down following the four year mark, but this is due to the fact that drives of this age level are not as common in the data. We hence normalize the number of swaps within a month by the amount of drives represented in the data at that month to produce an unbiased failure *rate* for each month (dashed line in Figure 6). We see that this rate evens out after the third month, indicating that the length of this observed high-failure infancy period is approximately 90 days. Accordingly, for the remainder of this paper, we distinguish drive swaps as *young* versus *old*, i.e., those swaps occurring before vs. after the 90-day mark. Beyond the 90-day mark, we observe that the failure rate is roughly constant, suggesting that, even as drives become very old, they are not any more prone to failure.

One potential explanation for the spike in failures for infant drives is that they are undergoing a "burn-in" period. This is a common practice in data centers, wherein new drives are subjected to a series of high-intensity workloads in order to test their resilience and check for manufacturing faults. These increased workloads could stress the drive, leading to a heightened rate of failure. To test this hypothesis, we looked at the intensity of workloads over time. For each month of drive age, we examined drives of that age and how many write operations they processed per day. The distributions of these write intensities are presented in Figure 7.

It is clear that younger drives do *not* tend to experience more write activity than usual (in fact, they tend to experience markedly *fewer* writes!). A similar trend is apparent for read activity (not pictured). We conclude that there is no burn-in period for these drives and that the spike in failure rates is caused by manufacturing malfunctions not caught by drive testing.

Beyond drive age, we are also interested in the relationship between failure and device wear, which we measure using P/E cycles, as discussed in Section 2. In the same style as Figure 6, Figure 8 illustrates the relationship of cumulative P/E cycles and probability of failure in the form of a CDF (solid line) and an accompanying failure rate (dashed line). The CDF illustrates that almost 98% of failures occur before the drive sees 1500 P/E cycles. This is surprising, considering that the manufacturer guarantees satisfactory drive performance up until 3000 P/E cycles. Conversely, the failure rate

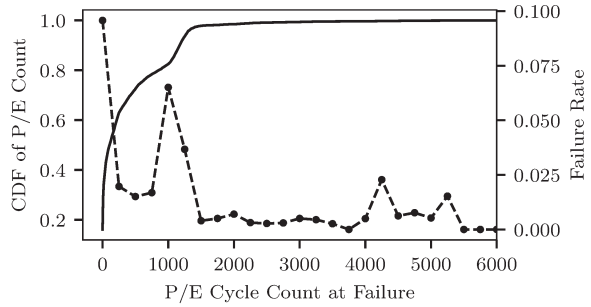

Figure 8: The distribution P/E cycle counts of failed drives (solid) and the proportion of drives that fail (dashed) at a given P/E level, binned in increments of 250 cycles.

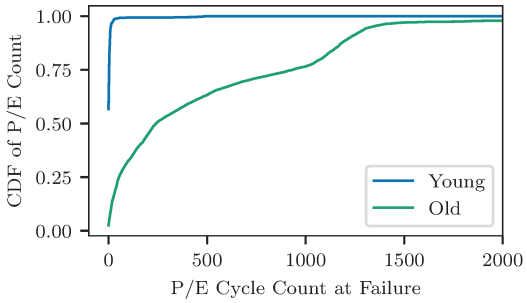

Figure 9: The CDF in Figure 8 split across infant failures (occurring at age $\leq$ 90 days) and mature failures (occurring at age $>$ 90 days).

beyond the P/E cycle limit is very small and roughly constant. The spikes at 4250 and 5250 P/E cycles are artifactual noise attributed to the fact that the number of drives that fail at these P/E levels are so few in number.

In the figures discussed, we observe high failures rates for both drives with age less than three months and drive with fewer than 250 P/E cycles. Due to the correlation between age and P/E cycles, these two characterizations may be roughly equivalent, describing the same phenomenon. However, we do not find this to be the case. To illustrate this, we plot two CDFs in Figure 9: one for young failures and one for old ones. It is clear that the young failures inhabit a distinct, small range of the P/E cycle distribution. Since this range is so small, the individual P/E cycle counts are not informative to young failures.

Observation #6: Age plays a crucial role in the SSD failure/swap incidence. In particular, drives younger than 90 days have markedly higher failure incidence rates. This phenomenon is characteristic to young drives and cannot be explained with P/E cycle counts.

Observation #7: Beyond the infancy period, age does not seem to have an important part to play in failure rate. The oldest drives seem to fail with roughly the same frequency as young, non-infant drives.

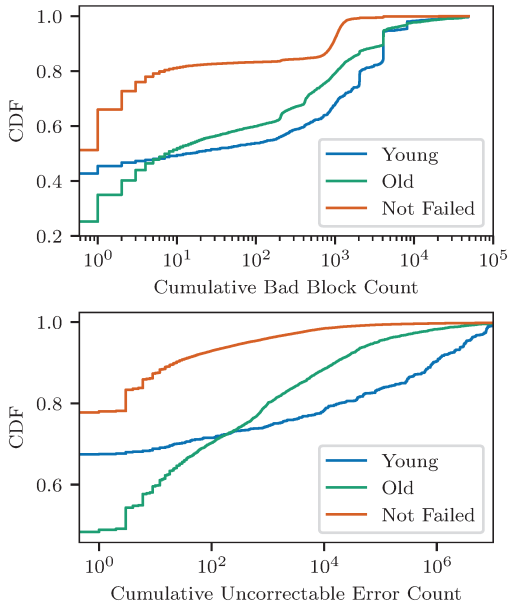

Figure 10: CDF of cumulative bad block counts and uncorrectable error counts based on the age at which the swap occurred. The "Not Failed" CDF corresponds to the distribution over drives that are not observed to fail.

Observation #8: The vast majority of drive failures happen well before the P/E cycle limit is reached. Furthermore, drives operating beyond their P/E cycle limit have very low rates of failure.

4.2 Error Incidence

Intuitively, we would expect that catastrophic drive failure is preceded by previous lapses in drive function, indicated in our data as non-transparent errors. In particular, we focus on uncorrectable errors and bad blocks since they are by far the most common of these errors. Other errors occur far too rarely to give much insight. We test the validity of our intuition by comparing the cumulative counts of errors seen by failed drives to a baseline of cumulative error counts taken across drives that are not observed to experience failure. We are also particularly interested to see if there is any difference in error incidence between young failures ($\leq$ 90 days) and old failures ($>$ 90 days). This is illustrated with CDFs in Figure 10.

We find that drives that fail tend to have experienced orders of magnitude more uncorrectable errors and bad blocks than we would expect, on average. This is exemplified by the fact that in roughly 80% of cases, non-failed drives are not observed to have experienced any uncorrectable errors. On the other hand, for failed drives, this proportion is substantially lower: 68% for young failures and 45% for old drives. In fact, broadening our scope, we find that 26% of failures happen to drives which have experienced no non-transparent errors *and* which have developed no bad blocks. Furthermore, we find that, if errors *are* observed, then young failures tend to see more of them than old failures. This is most easily seen in the tail behavior of the aforementioned CDFs; for example, the 90th percentile of the uncorrectable error count distribution is two orders of magnitude

larger for young failures than for old failures, in spite of the fact that the young drives have been in operation for much less time.

Overall, the presence of errors is not a very good indicator for drive failure since most failures occur without having seen *any* uncorrectable errors. However, drives that experience failure do have a higher rate of error incidence, which means that we expect error statistics to be of some utility in failure prediction (to be discussed in Section 5). Furthermore, we find that the patterns of error incidence are markedly different among young and old failures. In particular, young failures have a predilection toward extremely high error counts.

Moving into a finer temporal granularity, we are interested in error incidence directly preceding the failure. This behavior is of particular importance for failure forecasting and prediction. We ask: do drives tend to be more error-prone right before a failure occurs? How long before the drive failure is this behavior noticeable? Figure 11 shows two relevant uncorrectable error statistics in the period before a drive swap.

The top graph shows the probability that a faulty drive had an error within the last N days before its failure. The baseline is the probability of seeing an uncorrectable error within an arbitrary N-day period. We see that failed drives see uncorrectable errors with a much higher than average probability and that this behavior is most noticeable in the last two days preceding the failure. However, the probability that a failed drives does not to see any errors in its last 7 days is very high (about 75%).

The bottom graph shows the distribution of those uncorrectable error counts that are nonzero on each day preceding the swap. We find that error counts tend to increase as the failure approaches. We also find that young failed drives, if they suffer an error, tend to experience orders of magnitude more errors than older ones, note the log-scale on the y-axis of the graph.

To summarize, we zoom in specifically on the period directly preceding a failure. We show that error incidence rates depend on two factors: (1) the age of the drive (young vs. mature) and (2) the amount of time until the swap occurs. The resulting increase in error rate is most noticeable in the two days preceding the swap, suggesting that the lookahead window within which we can accurately forecast failure may be small.

Observation #9: Incidence of non-transparent errors is not strongly predictive of catastrophic drive failure. In fact, a substantial proportion of drives experience failure without having ever seen any sort of serious soft error.

Observation #10: Failures of young drives are more likely to have seen higher rates of error incidence than failures of mature drives.

Observation #11: Error incidence rates increase dramatically in the two days preceding a drive failure.

5 FAILURE PREDICTION

In order to shed light on the causes for failures, we have developed prediction models for the detection of swap-inducing failures. The models predict whether a failure will occur within the next N days for some $N \geq 1$.

The use case of prediction is clear: if we are able to detect future failures far enough in advance with sufficient certainty, we have the option to take preventative action to mitigate the risk of data loss and downtime. For example, this may involve backing up the faulty drive's data to another disk to act as a spare.

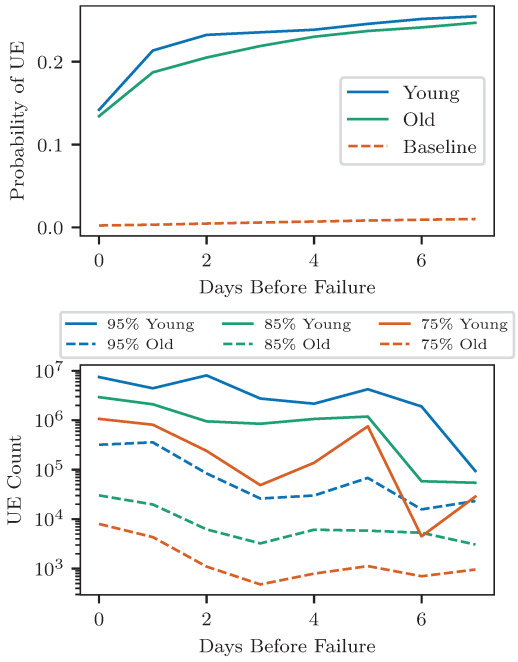

Figure 11: (Top) Probability of uncorrectable error (UE) happening within the last n days before a swap. The baseline curve on the top graph is the probability of seeing an uncorrectable error within an arbitrary n-day period. (Bottom) Provided a UE happens, how many occur? Upper percentiles of the distribution of uncorrectable error counts preceding to failure, excluding zero counts.

That being said, our goal in this section is not simply predictive accuracy. The machine learning classifiers we consider are simple yet powerful enough to facilitate interpretation and a more thorough investigation into the symptoms and causes of drive failure. For these classifiers, high predictive performance not only indicates the utility of the classifier in a predictive setting, but also its ability to model the log data. Accordingly, high model performance inspires trust in the inferences we draw from the model.

5.1 Model Description and Metrics

As **input**, we use each of the workload and error statistics itemized in Section 2. For each of these statistics, we include two values: the value of the statistic on the day of prediction as well as a cumulative count over the course of the drive's lifetime. Daily measures are used to provide information about current drive behavior while cumulative measures are used as they readily summarize all previous activity from across the drive's lifetime. For example, for each date of prediction, we include both the number of write operations performed by the drive on that day as well as the cumulative number of write operations performed, from the beginning of the drive's lifetime up to the date of prediction.

The **output** of the models that we consider is a continuous output in the interval [0, 1], which may be interpreted as the conditional probability of failure given the input. We often find it useful or necessary to have a binary prediction, i.e., failure vs. non-failure. Such binary predictions are of great practical importance. To accomplish this, we may discretize this output into a binary prediction using a discrimination threshold α: if the failure probability exceeds α, we predict that failure will occur, otherwise we predict that no failure will occur.

Consistent with the standard practice of evaluating a classifier with highly imbalanced data sets (as is the case here, our data contains 1 failure for each 10,000 non-failure cases), we choose to measure predictive performance using the receiver operating characteristic (ROC) curve [7] because it is insensitive to class imbalance. Furtheremore, the ROC curve illustrates the practical, diagnostic performance of a binary classifier as the discrimination threshold α is varied. The ROC curve plots the true positive rate (i.e., recall) against the false positive rate for the selected classifier. These are calculated as

$$\text{TPR} = \frac{\text{\# of True Positives}}{\text{\# of Positive Samples}}$$

$$\text{FPR} = \frac{\text{\# of False Positives}}{\text{\# of Negative Samples}}$$

The values of these two statistics vary depending on the chosen discrimination threshold α. Plotting a curve across all values of these two statistics, we find the area under this curve to obtain a summary statistic of classifier performance. This is called the ROC AUC (area under curve) statistic. The ROC AUC ranges between 0.5 (indicating performance not exceeding that of random guessing) and 1.0 (indicating perfect, deterministic prediction). We chose this metric since it is known to be robust in cases of imbalanced classes [7]. This is due to the fact that the true positive rate and false positive rate metrics are independent of the level of imbalance. The false negative rate for the classifier may also be read from the ROC curve using the identity FNR = 1 − TPR. This transformation allows a method of comparison between our predictions and those shown in a previous machine learning prediction study on this same data set that focuses on predicting the occurence of soft errors only [17].

For further assurance of model validity, 5-fold cross-validation is used. This is done by splitting the drive ID numbers into five equally sized groups. We train the model five times, successively using a different group as the testing set, with the remaining four groups used as the training set. For each train–test split, we calculate the ROC AUC value, and we report the mean value of the metric across the five splits. We found cross-validation to be important since the sampling bias for train–test split can be quite significant in this imbalanced classification problem. Furthermore, error and workload for a given drive are highly correlated across different drive days, leading to results that are biased to be larger than expected. Accordingly, we avoid splitting observations for a given drive across the training and testing sets. This is done by partitioning the folds based on drive ID.

Since we are dealing with a heavily imbalanced data set (out of 40,000,000 observed days of data, we have only 4000 failures), we randomly downsample the majority class to produce a 1:1 positive–negative ratio. This is a standard machine learning tactic used to make the classifier more sensitive to the minority class [13]. One concern with downsampling is that the model is not able to capture all of the variation within the majority class. To ensure the model's accuracy, we performed multiple random downsamplings on the same training set and observed the ROC AUC for each resulting model. We found that the AUC score only wavers on the order of ±0.001. We determined this variability induced by downsampling to be negligible. We also tested different downsampling ratios beyond 1:1 and observed either miniscule improvements or overall reductions in performance.

5.2 Prediction Accuracy

Table 6 reports on the average ROC AUC values for six machine learning predictors: logistic regression, k-nearest neighbors (k-NN), support vector machine (SVM), neural network, decision tree, and random forest [2]. For each method, we performed a grid search over hyperparameters in order to find the best configuration. Most of these hyperparameters were regularization parameters, tuning the complexity of the trained model. These include the ridge regression coefficient for logistic regression, the maximum depth of the trees in the random forest, and the sizes of the hidden layers in the neural network. We avoided overfitting by choosing the values of the hyperparameters that provided the best cross-validated performance with respect to ROC AUC. The classifiers (and respective AUC predictions) are for the entire log, i.e., we do not distinguish among drive types. Similarly to [17] where predictions of the disk error types is done (and not the catastrophic failures that we focus on here), we find that Random Forest models perform best on this data set compared to other common classifiers, including logistic regression and neural networks. We believe random forests to be so successful on this data set since they work well with discrete data are able to model nonlinear effects. It is also interesting to see that a single decision tree is able to achieve competitive performance to the random forest ensemble. In addition, across all models, it is clear that the shorter the lookahead window, the higher the quality of predictions we are able to achieve.

The performance of the Random Forest model is shown for a wider range of values for the detection window size N in Figure 12. We see that for lookahead that ranges up to 30 days, the effectiveness of prediction drops from 0.90 (1 day) to 0.77 (30 days), suggesting that strong prediction of swaps can be done with a range of windows but predictions are especially strong for 1 to 3 days lookahead.

Next, we consider the effectiveness of the classifier when evaluated individually on each drive type: MLC-A, MLC-B, and MLC-D. Figure 13 reports results on this for a 1-day lookahead window and show that the Random Forest model performs nearly identically across the three MLC logs. Next, to test robustness across MLC model types, we see whether predictive success on one model implies predictive success on another. To test this, we train the classifier using one MLC model in order to predict failures for another. Table 7 presents the ROC AUC results and shows that this is feasible with AUC values showing only minor degradation. Yet, if all data (all three MLC logs) are used for training (see the last column of Table 7), prediction is superior.

N (lookahead days)	1	2	3	7
Logistic Reg.	0.796 ± 0.010	0.765 ± 0.009	0.745 ± 0.007	0.713 ± 0.010
k-NN	0.816 ± 0.013	0.791 ± 0.009	0.772 ± 0.008	0.716 ± 0.008
SVM	0.821 ± 0.014	0.795 ± 0.011	0.778 ± 0.011	0.728 ± 0.011
Neural Network	0.857 ± 0.007	0.828 ± 0.004	0.803 ± 0.009	0.770 ± 0.008
Decision Tree	0.872 ± 0.007	0.840 ± 0.007	0.819 ± 0.005	0.780 ± 0.006
Random Forest	**0.905 ± 0.008**	**0.859 ± 0.007**	**0.839 ± 0.006**	**0.803 ± 0.008**

Table 6: ROC AUC for each prediction model and lookahead window N (in days). The cross-validated mean is presented with the standard deviation across folds. The AUC value of the best model is bolded.

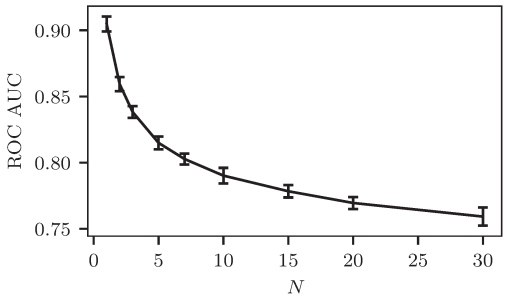

Figure 12: Random forest performance as a function of lookahead window size N. Error bars indicate the standard deviation of the cross-validated error across folds.

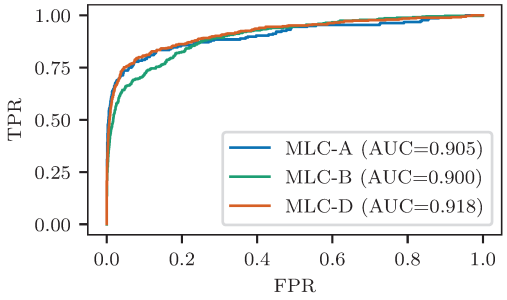

Figure 13: ROC curves for each drive model with a random forest prediction model and lookahead of $N = 1$ day.

	Training model(s)			
Test model	MLC-A	MLC-B	MLC-D	All
MLC-A	*0.891*	0.871	0.887	*0.901*
MLC-B	0.832	*0.892*	0.849	*0.893*
MLC-D	0.868	0.857	*0.897*	*0.901*

Table 7: Random forest for $N = 1$. Italics indicate that the AUC was estimated with cross-validation.

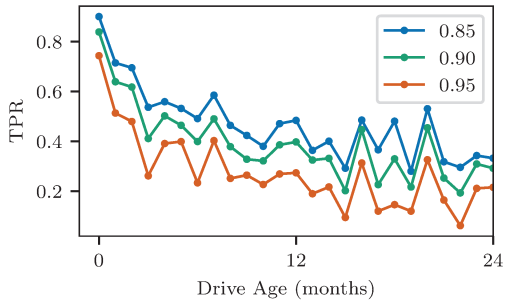

Figure 14: True positive rate as a function of drive ages. Three different prediction thresholds are reported.

5.3 Model Improvement: Using Drive Age

Recall from the previous section that we found that a large proportion of failures comes from infant drives (i.e., those less than 90 days old, labeled in our discussion as "young"). We may ask: are there differences in how the model detects infant failures versus mature failures? To see this, we evaluated model performance for each month of drive age. Since the random forest model outputs a prediction probability, to get a binary prediction, we threshold this output. A conservative threshold (i.e., a threshold close to 1) is practical for many real-world prediction applications since a low false positive rate is generally required. Using these binary predictions, we evaluated the true positive rate (i.e., recall) on the test set as a function of the age of the input drive. These rate estimates were 5-fold cross-validated and the mean TPR is reported. The resulting figure is shown in Figure 14. Three reasonable choices of probability thresholds are reported. We see that, for all three thresholds, the model is able to achieve significantly higher TPR for drives less than three months old. This is an indication that there are some aspects of infant failures which are more easily detected by the model, leading to superior performance during this period.

To further validate this distinction in predictive performance, we draw separate ROC curves for young and old drive inputs, much like we did for model type in Figure 13. This is shown in Figure 15. We confirm that the performance on young drives is consistently better than on older drives, as evidenced by the superior AUC score. To obtain even better performance, we chose to partition the data set completely into infant and mature age groups and train these two subsets separately. We find that infant drive classifier has a ROC

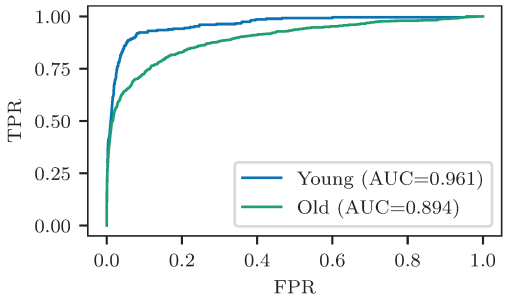

Figure 15: ROC curves for young (age at most 90 days) versus old (age more than 90 days) drives. Prediction model is random forest with lookahead $N = 1$ days.

AUC value of $\mathbf{0.970 \pm 0.005}$ while the classifier for older drives has an AUC that is only $\mathbf{0.890 \pm 0.005}$. Thus, young drive failures are fundamentally *more predictable* than those of older drives. This is significant because a large proportion (20%) of swap-inducing failures are young failures. We conclude that the indications of young drive failure must be more robust than—and fundamentally distinct from—the indications of mature drive failure. This result echoes our characterization of young and old failures discussed in Section 4.

5.4 Model Interpretability

The prediction results are respectable for such an imbalanced situation, but one of the great strengths of the Random Forest model is its interpretability. A trained Random Forest model can produce a ranking of input features based on what proportion of variance in the training set they are able to explain. The top features for the models for infant and mature drives are given in Figure 16. It is interesting to observe that the relative feature importance ranking across the two classifiers is indeed quite different. For infant drives, the age of drive is the most important one, followed by non-transparent errors: *cumulative* values of bad block counts, final read errors, read error counts, and uncorrectable error counts all have importance higher than than 0.08%. For mature drives, the features that are most useful for model prediction are instead those that relate to drive wear-and-tear, i.e., read counts, correctable error counts, and write counts, with the cumulative bad block count ranking as the fourth most important feature (for infant drives it is the most important feature). It is expected that read and write counts are important for failure prediction, given that a drive is more likely to not have any activity before a failure. So we conclude that the age and usage of the drive is a very important feature for model prediction accuracy.

The prediction scheme we discuss here bears similarity to previous results for this data set, which predicted uncorrectable error incidence [17]. Although we are predicting swap-inducing failures, our methodologies here are extensible to be able to improve the predictions of errors given in [17]. For example, by partitioning the data into young and old sections, we are able to achieve ROC AUC scores of 0.960 ± 0.009 and 0.931 ± 0.007, respectively. This leads to improved performance when compared to the AUC score of 0.933 ± 0.006 obtained when training on unpartitioned data. We see

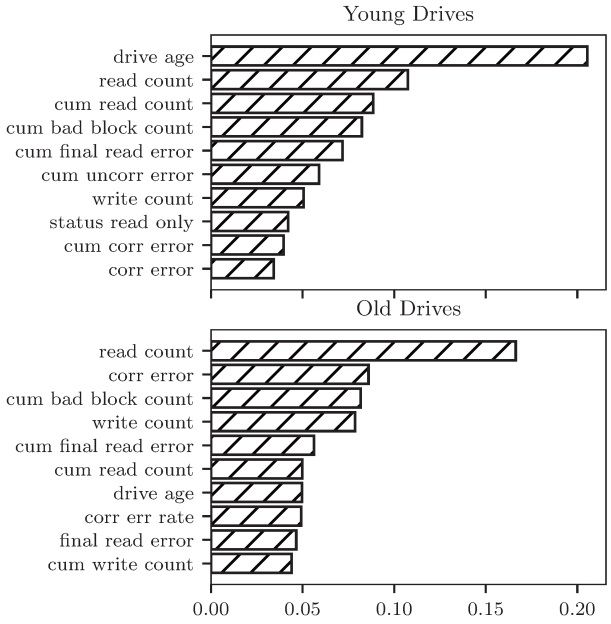

Figure 16: Feature importance for the random forest models for infant drives (top figure) and mature drives (bottom).

this same pattern for the prediction of a variety of non-transparent errors in Table 8.

Observation #12: The features that are most important for prediction are very different between young and old failures. In particular, young failures are dominated by age and non-transparent errors, while old failures are more related with drive wear-and-tear.

Observation #13: Infant drive failures are more easily predicted than mature drive failures, indicating that the symptoms are infant drive failures are more robust. This trend holds when predicting a variety of non-transparent errors, as well. Training separate models leads to a substantial boost in predictive performance for young failures.

6 RELATED WORK

Large-scale data centers from major service priviers are extensively used to serve millions of jobs on a daily basis [8, 28, 29]. Consequently, reliability of such systems is of great importance. Much prior work has investigated and analyzed the impact of failures and errors on large-scale data centers [11, 16, 21, 26, 30]. The storage system hierarchy has been singled out as especially important for reliability, especially for DRAM and HDD failures [14, 24, 27]. In particular, the other large-scale reliability study of SSDs coming from Facebook [18] focuses more on hardware-level events. We echo some of their conclusions regarding infant failures in this paper.

In order to initiate proactive actions (e.g., drive replacement) before the failure occurs, many works focus on predicting system failures [12, 16, 22]. Hamerly et al. [12] use Bayesian approaches while Ma et al. [16] simply use threshold-based prediction. Machine learning based failure prediction shows more advantages

Error	Combined	Young	Old
Bad block	0.877	0.878	0.873
Erase	0.889	0.934	0.882
Final read	0.906	0.959	0.852
Final write	0.841	0.937	0.780
Meta	0.854	0.890	0.842
Read	0.971	0.917	0.973
Response	0.806	—	—
Timeout	0.755	0.812	0.735
Uncorrectable	0.933	0.960	0.931
Write	0.916	0.911	0.914

Table 8: ROC AUCs for random forest to predict various error types for $N = 2$. Response errors are too rare to predict for different age granularities.

than heuristics and statistical models. Mahdisoltani et al. [17] explore different machine learning models to predict uncorrectable errors and bad blocks in hard disk drives and solid state disks. Ding et al. [6] capture the fuzzy rules and combine time series models to predict online software system failures. Botezatu et. al. [3] and Narayanan et. al. [20] use machine learning to predict disk replacements using SMART data. Xu et. al. [31] use SMART data and system-level signals to develop a machine learning model to improve service availability of Microsoft Azure.

Compared with statistical models in prior works [12, 16, 22], our work leverages machine learning models to predict disk drive failures with low false positive rate. In contract to prediction for systems in a manufacturer-controlled environment [19] or using data sets of very limited size [9], we focus here on trace data collected in a production system consisting of over 10,000 drives over a period of six years. The same trace has been examined in [17] but focused on machine learning models for predicting SSD errors rather than complete disk failures. We recreate and expand their work here in Section 5.4. Schroeder et al. [25] also utilized the same trace, but focused orthogonally on bit error rates, without discussing causes, symptoms, or prediction.

7 CONCLUSION

In this paper we do a detailed workload characterization study of SSD failures using a Google trace of more than 30,000 drives within a time period of six years in production data centers. We reach several surprising conclusions, in particular, that the usual suspects of drive failure (write behavior and error incidence) are nowhere near as informative as one would expect. We extract informative features from this characterization to train several machine learning predictors, and find that random forests are the most successful, achieving extremely accurate predictions of drive failure. Our analysis concludes that the age of the drive is a crucial factor for failure prediction. If the drive does not fail within its first 3 months of operation, then wear and tear play a more substantial role in its reliability. We are currently working on advancing our understanding of disk activity prior to a swap and directly following re-entry in order to improve our prediction models for large N.

ACKNOWLEDGMENT

We deeply thank Arif Merchant of Google for making the SSD trace available to us. The authors would also like to thank the anonymous reviewers for their comments and suggestions. This material is based upon work supported by the National Science Foundation (NSF) grants (#1717532 and #CCF-1649087). This work was performed in part using computing facilities at the College of William & Mary which were provided by contributions from the NSF and the Commonwealth of Virginia Equipment Trust Fund.

REFERENCES

[1] American National Standards Institute. AT attachment 8 - ATA/ATAPI command set (ATA8-ACS), 2008. http://www.t13.org/documents/uploadeddocuments/docs2008/d1699r6a-ata8-acs.pdf. (????).
[2] Ethem Alpaydin. 2014. *Introduction to machine learning.* MIT press.
[3] Mirela Madalina Botezatu, Ioana Giurgiu, Jasmina Bogojeska, and Dorothea Wiesmann. 2016. Predicting Disk Replacement towards Reliable Data Centers. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016.* 39–48.
[4] Yu Cai, Yixin Luo, Erich F. Haratsch, Ken Mai, and Onur Mutlu. 2015. Data retention in MLC NAND flash memory: Characterization, optimization, and recovery. In *21st IEEE International Symposium on High Performance Computer Architecture, HPCA 2015, Burlingame, CA, USA, February 7-11, 2015.* 551–563.
[5] G.W. Corder and D.I. Foreman. 2014. *Nonparametric Statistics: A Step-by-Step Approach.* Wiley.
[6] Zuohua Ding, Yuan Zhou, Geguang Pu, and MengChu Zhou. 2018. Online Failure Prediction for Railway Transportation Systems Based on Fuzzy Rules and Data Analysis. *IEEE Transactions on Reliability* (2018).
[7] Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern Recognition Letters* 27, 8 (2006), 861 – 874. DOI: https://doi.org/10.1016/j.patrec.2005.10.010 ROC Analysis in Pattern Recognition.
[8] Phillipa Gill, Navendu Jain, and Nachiappan Nagappan. 2011. Understanding network failures in data centers: measurement, analysis, and implications. In *ACM SIGCOMM Computer Communication Review*, Vol. 41. ACM, 350–361.
[9] Moises Goldszmidt. 2012. Finding Soon-to-Fail Disks in a Haystack.. In *HotStorage*.
[10] Laura M. Grupp, Adrian M. Caulfield, Joel Coburn, Steven Swanson, Eitan Yaakobi, Paul H. Siegel, and Jack K. Wolf. 2009. Characterizing flash memory: anomalies, observations, and applications. In *42st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-42 2009), December 12-16, 2009, New York, New York, USA.* 24–33.
[11] Chuanxiong Guo, Lihua Yuan, Dong Xiang, Yingnong Dang, Ray Huang, Dave Maltz, Zhaoyi Liu, Vin Wang, Bin Pang, Hua Chen, and others. 2015. Pingmesh: A large-scale system for data center network latency measurement and analysis. In *ACM SIGCOMM Computer Communication Review*, Vol. 45. ACM, 139–152.
[12] Greg Hamerly, Charles Elkan, and others. 2001. Bayesian approaches to failure prediction for disk drives. In *ICML*, Vol. 1. 202–209.
[13] Haibo He and Edwardo A Garcia. 2008. Learning from imbalanced data. *IEEE Transactions on Knowledge & Data Engineering* 9 (2008), 1263–1284.
[14] Andy A Hwang, Ioan A Stefanovici, and Bianca Schroeder. 2012. Cosmic rays don't strike twice: understanding the nature of DRAM errors and the implications for system design. In *ACM SIGPLAN Notices*, Vol. 47. ACM, 111–122.
[15] Yixin Luo, Saugata Ghose, Yu Cai, Erich F. Haratsch, and Onur Mutlu. 2018. Improving 3D NAND Flash Memory Lifetime by Tolerating Early Retention Loss and Process Variation. In *Abstracts of the 2018 ACM International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS 2018, Irvine, CA, USA, June 18-22, 2018.* 106.
[16] Ao Ma, Rachel Traylor, Fred Douglis, Mark Chamness, Guanlin Lu, Darren Sawyer, Surendar Chandra, and Windsor Hsu. 2015. RAIDShield: characterizing, monitoring, and proactively protecting against disk failures. *ACM Transactions on Storage (TOS)* 11, 4 (2015), 17.
[17] Farzaneh Mahdisoltani, Ioan Stefanovici, and Bianca Schroeder. 2017. Proactive error prediction to improve storage system reliability. In *2017 USENIX Annual Technical Conference (USENIX ATC 17). Santa Clara, CA: USENIX Association.* 391–402.
[18] Justin Meza, Qiang Wu, Sanjeev Kumar, and Onur Mutlu. 2015. A Large-Scale Study of Flash Memory Failures in the Field. In *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, Portland, OR, USA, June 15-19, 2015.* 177–190.
[19] Joseph F Murray, Gordon F Hughes, and Kenneth Kreutz-Delgado. 2005. Machine learning methods for predicting failures in hard drives: A multiple-instance application. *Journal of Machine Learning Research* 6, May (2005), 783–816.

[20] Iyswarya Narayanan, Di Wang, Myeongjae Jeon, Bikash Sharma, Laura Caulfield, Anand Sivasubramaniam, Ben Cutler, Jie Liu, Badriddine M. Khessib, and Kushagra Vaid. 2016. SSD Failures in Datacenters: What? When? and Why?. In *Proceedings of the 9th ACM International on Systems and Storage Conference, SYSTOR 2016, Haifa, Israel, June 6-8, 2016*. 7:1–7:11.

[21] Bin Nie, Ji Xue, Saurabh Gupta, Tirthak Patel, Christian Engelmann, Evgenia Smirni, and Devesh Tiwari. 2018. Machine Learning Models for GPU Error Prediction in a Large Scale HPC System. In *Dependable Systems and Networks (DSN), 2018 48th Annual IEEE/IFIP International Conference on*. IEEE, 1–12.

[22] Eduardo Pinheiro, Wolf-Dietrich Weber, and Luiz André Barroso. 2007. Failure Trends in a Large Disk Drive Population.. In *FAST*, Vol. 7. 17–23.

[23] Moinuddin K. Qureshi, Dae-Hyun Kim, Samira Manabi Khan, Prashant J. Nair, and Onur Mutlu. 2015. AVATAR: A Variable-Retention-Time (VRT) Aware Refresh for DRAM Systems. In *45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2015, Rio de Janeiro, Brazil, June 22-25, 2015*. 427–437.

[24] Bianca Schroeder and Garth A Gibson. 2007. Disk failures in the real world: What does an MTTF of 1, 000, 000 hours mean to you?. In *FAST*, Vol. 7. 1–16.

[25] Bianca Schroeder, Raghav Lagisetty, and Arif Merchant. 2016. Flash Reliability in Production: the Expected and the Unexpected. In *14th USENIX Conference on File and Storage Technologies, FAST 2016, Santa Clara, CA, USA, February 22-25, 2016*. 67–80.

[26] Bianca Schroeder, Arif Merchant, and Raghav Lagisetty. 2017. Reliability of NAND-based SSDs: What field studies tell us. *Proc. IEEE* 105, 9 (2017), 1751–1769.

[27] Bianca Schroeder, Eduardo Pinheiro, and Wolf-Dietrich Weber. 2009. DRAM errors in the wild: a large-scale field study. In *ACM SIGMETRICS Performance Evaluation Review*, Vol. 37. ACM, 193–204.

[28] Abhishek Verma, Luis Pedrosa, Madhukar Korupolu, David Oppenheimer, Eric Tune, and John Wilkes. 2015. Large-scale cluster management at Google with Borg. In *Proceedings of the Tenth European Conference on Computer Systems*. ACM, 18.

[29] Guohui Wang and TS Eugene Ng. 2010. The impact of virtualization on network performance of Amazon EC2 data center. In *Infocom, 2010 proceedings ieee*. IEEE, 1–9.

[30] Guosai Wang, Lifei Zhang, and Wei Xu. 2017. What Can We Learn from Four Years of Data Center Hardware Failures?. In *Dependable Systems and Networks (DSN), 2017 47th Annual IEEE/IFIP International Conference on*. IEEE, 25–36.

[31] Yong Xu, Kaixin Sui, Randolph Yao, Hongyu Zhang, Qingwei Lin, Yingnong Dang, Peng Li, Keceng Jiang, Wenchi Zhang, Jian-Guang Lou, Murali Chintalapati, and Dongmei Zhang. 2018. Improving Service Availability of Cloud Systems by Predicting Disk Error. In *2018 USENIX Annual Technical Conference, USENIX ATC 2018, Boston, MA, USA, July 11-13, 2018*. 481–494.

[32] Mai Zheng, Joseph Tucek, Feng Qin, and Mark Lillibridge. 2013. Understanding the robustness of SSDs under power fault. In *Proceedings of the 11th USENIX conference on File and Storage Technologies, FAST 2013, San Jose, CA, USA, February 12-15, 2013*. 271–284.