Proactive error prediction to improve storage system reliability

Farzaneh Mahdisoltani, ***University of Toronto;*** **Ioan Stefanovici,** ***Microsoft Research;***
Bianca Schroeder, ***University of Toronto***

https://www.usenix.org/conference/atc17/technical-sessions/presentation/mahdisoltani

This paper is included in the Proceedings of the 2017 USENIX Annual Technical Conference (USENIX ATC '17).

July 12–14, 2017 • Santa Clara, CA, USA

ISBN 978-1-931971-38-6

Open access to the Proceedings of the 2017 USENIX Annual Technical Conference is sponsored by USENIX.

Improving Storage System Reliability with Proactive Error Prediction

Farzaneh Mahdisoltani
University of Toronto

Ioan Stefanovici
Microsoft Research

Bianca Schroeder
University of Toronto

Abstract

This paper proposes the use of machine learning techniques to make storage systems more reliable in the face of sector errors. Sector errors are partial drive failures, where individual sectors on a drive become unavailable, and occur at a high rate in both hard disk drives and solid state drives. The data in the affected sectors can only be recovered through redundancy in the system (e.g. another drive in the same RAID) and is lost if the error is encountered while the system operates in degraded mode, e.g. during RAID reconstruction.

In this paper, we explore a range of different machine learning techniques and show that sector errors can be predicted ahead of time with high accuracy. Prediction is robust, even when only little training data or only training data for a different drive model is available. We also discuss a number of possible use cases for improving storage system reliability through the use of sector error predictors. We evaluate one such use case in detail: We show that the mean time to detecting errors (and hence the window of vulnerability to data loss) can be greatly reduced by adapting the speed of a scrubber based on error predictions.

1 Introduction

While the storage landscape has changed significantly over the last decade, with hard disk drives (HDDs) and solid state drives (SSDs) each accounting for large shares of the market for persistent storage, one of the key requirements for storage systems has remained the same: store data reliably.

In addition to whole-drive failure, where a drive stops functioning in a way that necessitates replacement, a major threat to storage reliability are *partial* drive failures, where individual sectors on a drive cannot be read. This happens, for example when data in the affected sector is too corrupted to be corrected by drive-internal error correcting codes (ECC). For hard disk drives it can also be due to mechanical damage on the disk surface. The result is the same in either case: the drive cannot recover the data previously stored in the sector.

Field studies show that both solid state drives and hard disk drives experience sector errors at a significant rate. Recent studies based on data from Facebook and Google report that 20-57% of solid state drives experience at least one sector error [15, 22]. A 10-year old study by Bairavasundaram et al. [4] reports that 5–20% of nearline hard disk drives in Netapp's storage systems develop sector errors over a period of 24 months. Our own analysis of recent field data in this paper finds two drive models among the seven most common hard disk models in a large production installation with 11% and 25% of drives affected by sector errors, respectively (see Table 1).

Sector errors are a major concern also when looking into the future. Both SSDs and HDDs continuously grow in capacity to keep up with consumer demand. As there are more sectors on a drive, there is a larger chance that sectors will fail, and as capacity increases, so do drive densities, which can further increase the chances for bit corruption. And some data center operators argue that in order to continue to produce drives at an acceptable price point and with the desired performance characteristics, data center drives should be allowed to have higher error rates and responsibility to deal with those errors should be shifted to higher layers in the storage system [5].

The nature of sector errors makes them challenging to protect against, as they are *latent* errors, i.e. the drive is not aware of and will not report these errors until the affected sector is being accessed. That means storage systems need to proactively periodically read and verify data (a process called disk scrubbing), in order to avoid a situation where a sector error is discovered at a time when it cannot be recovered via redundancy in the system (e.g. during RAID reconstruction).

In this paper we make the case that storage systems would be better prepared to handle sector errors, if errors were predictable. We present techniques for accurately

Drive model	Capacity (TB)	#Drives	Drives (Drive days) affected by:			
			SMART 5	SMART 187	SMART 196	SMART 197
Seagate ST4000DM000	4	36368	1.19% (0.02%)	2.33% (0.01%)	N/A	3.37% (0.02%)
Hitachi HDS5C3030ALA630	3	4664	3.58% (0.05%)	N/A	2.55% (0.04%)	2.72% (0.01%)
HGST HMS5C4040ALE640	3	7168	0.91% (0.03%)	N/A	0.91% (0.03%)	0.59% (0.002%)
Hitachi HDS722020ALA330	2	4774	11.84% (0.12%)	N/A	9.76% (0.08%)	6.47% (0.03%)
HGST HMS5C4040BLE640	4	9426	0.24% (0.003%)	N/A	0.24% (0.003%)	0.32% (0.002%)
Hitachi HDS5C4040ALE630	4	2719	2.54% (0.03%)	N/A	1.62% (0.02%)	1.95% (0.005%)
Seagate ST3000DM001	3	4707	25.15% (1.77%)	30.59% (0.31%)	N/A	35.33% (0.29%)

Table 1: *Overview of HDD models*

predicting errors and show through one specific use case how these predictors can be used to improve storage system reliability. More precisely, we are making the following two contributions:

• We explore a variety of machine learning techniques and show that machine learning models can be trained to predict sector errors with high accuracy. Interestingly, we observe that some of the simplest and easiest to train machine learning models, random forests, are among the most accurate predictors. We also find that the training of predictors is robust, in that even smaller training data sets are sufficient for successful training, and that predictors trained on one drive model can be used to predict errors on a different drive model.

• We propose a number of different use cases for error predictors and explore one of them in depth: improving storage system reliability by adjusting scrub rates based on error prediction. Currently most storage systems run a background scrubber, which at a slow constant speed reads and verifies the stored data to proactively detect errors. Setting the right speed at which to perform scrubbing is tricky, as an overly aggressive scrubber can impact the performance of concurrently running foreground workloads, while a slow scrub speed increases the time it takes to detect an error, hence increasing the system's window of vulnerability to data loss. We propose to adjust scrub speed based on an error predictor, scrubbing faster when errors are predicted and more slowly otherwise. We show that by adjusting the scrub speed based on a predictor the window of vulnerability can be shortened by nearly a factor of 2X, while using the accelerated scrub rate for less than 2% of the total time.

2 A Look at the Field Data

We begin with a description of the field data that our study is based on, including a description of the various error modes and some summary statistics on error frequencies.

2.1 Hard Disk Drives

For our study of errors on hard disk drives and their prediction we use data that has been made publicly available by Backblaze [3]. The entire dataset covers more than a dozen different HDD models and more than a billion device hours, but some models have very small populations. Table 1 provides some summary statistics on the seven drive models with the most data.

The data for these drives is based on SMART (Self-Monitoring, Analysis and Reporting Technology). SMART is a monitoring system supported by most drives that reports on various indicators of drive health, including various types of errors, but also operational data, such as drive temperature, and power on hours of the drive. Backblaze collects for each drive daily snapshots of all SMART values reported by the drive.

To gauge how frequent sector errors are in the drive population at Backblaze, we consider the following SMART parameters that are related to sector errors. Note that the exact definition and reporting of these parameters varies between drive models and manufacturers, and that not all parameters are reported by all drive models.

SMART 5: Count of reallocated sectors. When a read or a write operation on a sector fails, the drive will mark the sector as bad and remap (reallocate) it to a spare sector on disk.

SMART 187: The number of read errors that could not be recovered using hardware ECC.

SMART 196: The total count of attempts to transfer data from reallocated sectors to a spare area. Unsuccessful attempts are counted as well as successful.

SMART 197: Count of "unstable" sectors. Some drives mark a sector as "unstable" following a failed read, and remap it only after waiting for a while to see whether the data can be recovered in a subsequent read or when it gets overwritten.

We begin by asking how common sector errors are on the Backblaze drives, since the most recent numbers in the literature [4] are based on data collected more than a decade ago. Table 1 shows, for the most common drive models at Backblaze, the fraction of disk drives and the fraction of drive days that are affected by any of the events corresponding to the 5 SMART parameters above.

We observe that the fraction of drives affected by sector errors is significant: for two of the models 11% and 25%, respectively, of their population have experienced

SSD Model	#Drives	Capacity	Lithography (nm)	PE cycle limit	Avg. PE cycles	Drives (Drive weeks) affected by:	
						Uncorrectable Errors	Bad Blocks
MLC-A	10115	480GB	50	3,000	730	37.07% (0.63%)	50.35% (0.44%)
MLC-B	10151	480GB	43	3,000	949	65.56% (1.09%)	82.75% (1.59%)
MLC-D	10258	480GB	50	3,000	544	46.72% (0.89%)	55.01% (0.57%)

Table 2: *Overview of SSD models*

at least one reallocated sector. We also note that these numbers are significantly higher than the averages reported in previous work [4], which was based on data collected in 2004-2006 in Netapp storage systems and saw 3.45% of drives affected by latent sector errors. However, the numbers we see are in line with those reported for the three nearline drives in the Netapp study, which ranged from 5–20%.

2.2 Solid State Drives

We have been able to obtain data for a randomly sampled subset of around 30,000 drives from three of the four MLC drive models used in a recent field study [22] on SSD reliability based on drives in Google's data centers. We refer to the models as MLC-A, MLC-B, and MLC-D, keeping the naming consistent with that in [22].

The drives in the dataset are based on commodity MLC flash chips, but are custom designed using a custom PCIe interface, firmware and driver. As such, reporting and monitoring is also customized (rather than relying on SMART). For each drive the data contains daily counts for a variety of different types of errors, as well as workload statistics, such as the number of read, write and erase operations during that day. Table 2 summarizes the key statistics for the drives in our data set.

The two events that we are most interested in are uncorrectable errors and bad blocks:

Uncorrectable errors (UEs): A read operations encounters more corrupted bits than the drive-internal ECC can correct. The drive returns an error.

Bad blocks: The drives at Google declare a block bad after an uncorrectable read error, a write error or an erase error, and consequently remap it (i.e. it is removed from future usage and any data that might still be on it and can be recovered is remapped to a different block). Unlike bad blocks for hard disk drives, which refer to disk sectors (typically 512 or 4096 bytes), the blocks here are the unit at which the SSD performs erase operations. The size of an erase block varies with the drive model, but is typically on the order of hundreds of KBs.

3 Predicting Errors

3.1 A Machine Learning Formulation

Our goal is to predict whether a drive will have a sector error within a given time interval, based on its past behavior, as captured by the monitoring data that it reported. We formulate the problem of predicting future errors as a classification problem and then use a variety of methods from machine learning to train classifiers. For simplicity, we assume in the discussion below that we are predicting errors one week into the future, i.e. whether there will be an error within the next 7 days. We create instances (or observations) to our classifier by dividing the data into non-overlapping one-week intervals. For each one-week interval, the response variable (to be predicted) is binary, set either to error or no-error, depending on whether a sector error was observed during this week or not. For explanatory variables (features) we consider all parameters that the drive reports as candidates. The explanatory variables, are based on the monitoring data that the drive produced prior to the prediction interval. More details on the setup of the machine learning formulation follow below.

3.1.1 The Response Variables

As explained in Section 2.1 there are a number of SMART parameters related to sector errors; their exact interpretation can vary between models and not all parameters are reported by all models. We believe that SMART 5 (S5) is the most interesting choice as a response variable, as it is consistently reported by all drive models and because it comprises all the different scenarios that might have led to declaring a sector bad (e.g. independently of how the bad sector was discovered), as it refers to the total number of sectors that have been reallocated. However, we also experiment with training classifiers to predict S187 (read errors that could not be recovered using ECC) and S197 (sectors became unstable).

For the SSD data choosing the response variables is straightforward, since all drive models use the same customized reporting mechanisms. We experiment with two different response variables: uncorrectable errors and bad blocks.

3.1.2 The Explanatory Variables

For the HDD data we consider all SMART parameters reported by a drive as possible candidates for explanatory variables. We convert the raw data into explanatory variables in two ways. The first set of input variables consists of the most recent raw values of all the parame-

ID#	Attribute Name
*S*1	Read Error Rate
*S*4	Start/Stop Count
*S*5	Reallocated Sectors Count
*S*7	Seek Error Rate
*S*9	Power-On Hours (POH)
*S*12	Power Cycle Count
*S*187	Reported Uncorrectable Errors
*S*193	Load Cycle Count
*S*194	Temperature
*S*197	Current Pending Sector Count
*S*199	UltraDMA CRC Error Count

Table 3: *SMART attributes selected as learning features for HDD devices*

ters reported by the drive before the beginning of the one-week interval. We also add a second set of input variables based on transformations of some smart parameters. In particular, some SMART parameters are reported as cumulative counts over a drive's lifetime (e.g. SMART 189 contains the total number of high fly writes a disk has ever experienced). However, for predicting errors during a given time period, it might be more insightful to know the recent rate of change for this variable, rather than the cumulative lifetime count. We therefore include for each cumulative smart parameter a second input variable that consists of the *increase* in value that the corresponding smart parameter experienced during the one-week window into the past.

As the resulting number of input variables is large, we perform feature selection before training machine learning models on the data. We use correlation coefficients and information gain to determine the features that correlate most with the drive errors and we remove the SMART attributes that never change for a specific drive model. Table 3 lists the SMART attributes we used to build prediction models for HDDs. For a more detailed explanation of the various fields see an overview of all SMART parameters [23]. Table 4 provides the attributes used for SSDs. For a more detailed explanation of the various fields, see [22].

3.1.3 Training the Classifiers

We experiment with five different machine learning methods that are commonly used for classification problems: classification and regression trees (CART), random forests, support vector machines, neural networks and logistic regression. For random forests we experimented with different numbers of trees, and settled on using 20 trees for the results included in the paper. We ran experiments with up to 100 trees, but did not see significant improvements. For support vector machines, we experiment with three different kernels: polynomial, linear and radial basis function (RBF) kernels. We also experimented with different degrees for the polynomial kernels. For neural networks, we include results for a network with 3 layers and 100 nodes. Neural networks with larger numbers of layers are impractical for learning rare events (such as errors or failures) as they require massive amounts of training data. We also experimented with more advanced type of neural networks, such as recurrent neural networks, but didn't find the results to improve upon standard neural networks, and hence chose not to include the results. We use the hold-out method to find the best values to adapt the parameters of neural networks, including learning rate, momentum and regularization factors. We perform a grid search on these parameters to find the combination that achieves the highest performance. For logistic regression we experimented with different values for regularization and learning rate. All methods were implemented in Matlab. For SVM we used the LIBSVM library [6].

ID#	Attribute Name
1	correctable error
2	erase count
3	erase error
4	factory bad block
5	final read error
6	final write error
7	meta error
8	read count
9	read error
10	response error
11	status dead
12	status read only
13	timeout error
14	timestamp usec
15	uncorrectable error
16	write count
17	write error
18	cumulative bad block count fixed
19	weekly bad block count
20	cumulative pe cycle fixed
21	weekly pe cycle

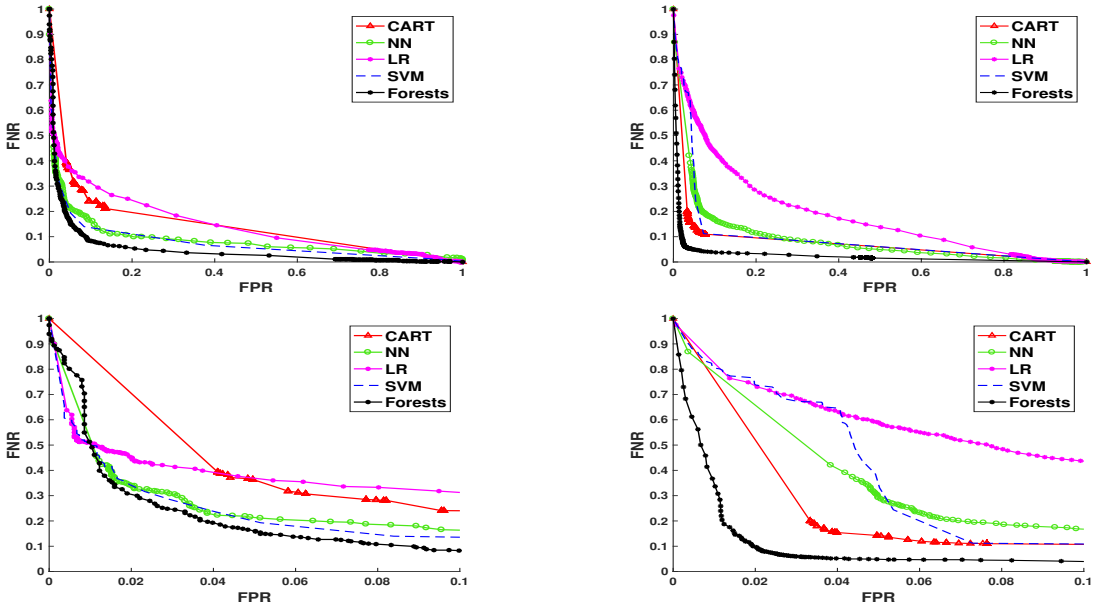
Table 4: *Attributes selected as learning features for SSD devices*

As the range of values spanned by different features varies widely, we employed data normalization using the feature scaling method to avoid bias towards features with larger parameter values. Feature scaling transforms each attribute in the data using the following formula:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

where X is the original value of a feature. X_{max} and X_{min} are respectively the maximum value and the minimum value of this feature for the subset of data for each disk manufacturer and model.

When creating the training data sets, we use majority class under-sampling, a standard technique to improve training in the case of very imbalanced classes, which arises because the original data set has many more in-



(a) Hitachi (b) Seagate

Figure 1: *False positive rates (x-axis) versus false negative rates (y-axis) when predicting sector reallocation (SMART 5) for Hitachi and Seagate HDDs. The bottom row shows a close-up of the false positive range* $[0;0.1]$ *on the x-axis.*

stances of negatives (no error), than positives (error). E.g. if we randomly selected training instances from the entire data set, the training set would include only a very small number of instances with errors and hence bias the training process towards non-error predictions. Instead we undersample with different ratios, making sure that at least a certain fraction of training instances are error instances. We experimented with different ratios ranging from 1:1 to 1:10, and found that most ratios, where about 20-60% of all training instances are positives, work well. We have chosen to include the results for a ratio of 1:3.

When performing the training, we divide the data into 75% training data and 25% testing data for most experiments, as is common practice in the machine learning community. Later in the paper, we will also show (Section 3.3) that much smaller training sets are sufficient. We used the hold-out method for tuning parameters of different algorithms, and chose the values which led to the highest quality predictions.

3.1.4 Metrics

We measure the success of different machine learning models by reporting two standard measures: the false positive rate (FPR) and the false negative rate (FNR).

The false positive rate measures the frequency of false alarms. It is the fraction of time intervals that did not experience an error, but was falsely predicted to have an error:

$$\text{FPR} = \frac{\#\text{false positives}}{\#\text{false positives} + \#\text{true negatives}}$$

The false negative rate measures what fraction of errors was missed, i.e. the fraction of intervals that had an error, but was predicted not to have an error:

$$\text{FNR} = \frac{\#\text{false negatives}}{\#\text{false negatives} + \#\text{true positives}}$$

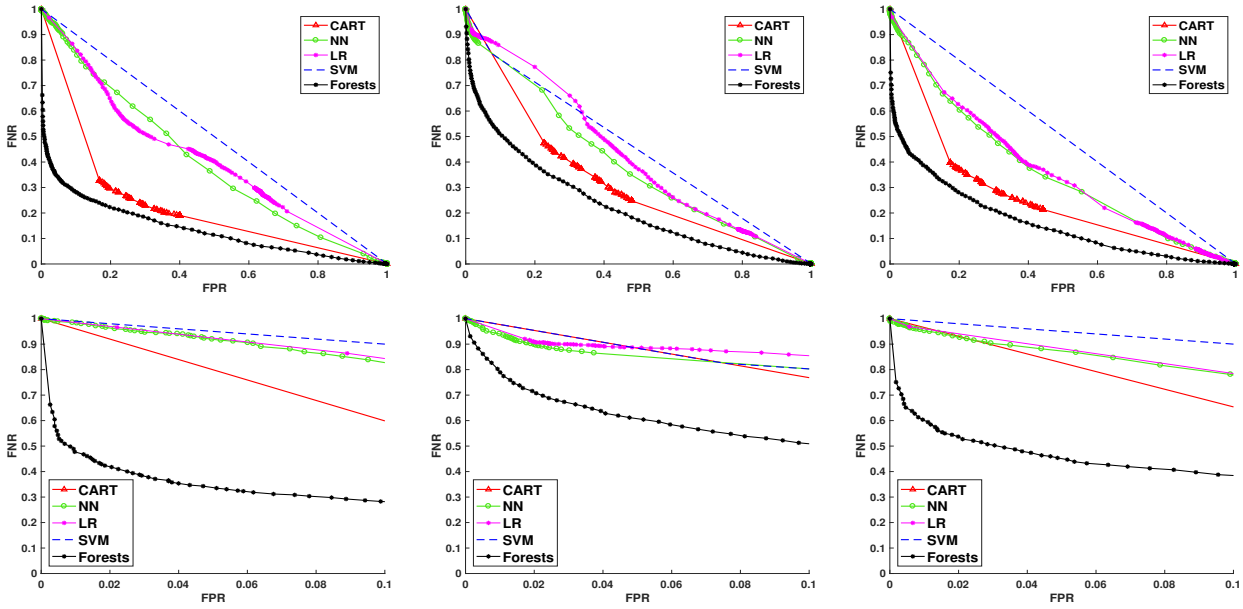
3.2 Prediction Results

3.2.1 Prediction Results for HDDs

We train classifiers to predict SMART 5 (sector reallocations) for two of the hard disk drive models, Hitachi's HDS722020ALA330 and Seagate's ST3000DM001. We chose those two drive models to cover two different manufacturers, because they are among the most common drives in the HDD dataset and because they are the drive models with the highest error rates.

The graphs in Figure 1 summarize the quality of the predictions we obtain using the various machine learning methods on each of the two device types. The top row shows results for the entire range of false positive ratios (x-axis), while the bottom row shows a close-up of the x-axis range with false positive ratios less than 0.1.

We make several observations. First, errors can be predicted with a high accuracy. For example, when limiting the false positive rate (i.e. the rate of false alarms) to 10% we can correctly predict 90% and 95% of all errors for Hitachi and Seagate, respectively. When limiting the false positive rate more conservatively to 2% we can still correctly predict 70-90% of the errors.



(a) MLC-A (b) MLC-B (c) MLC-D

Figure 2: *False positive rates (x-axis) versus false negative rates (y-axis) when predicting uncorrectable errors for the three SSD models. The bottom row shows a close-up of the false positive range* $[0;0.1]$ *on the x-axis.*

To put those false positive rates into context, recall that our goal is to use predictions to proactively trigger light-weight protection mechanisms, such as a data scrub to proactively detect errors. So, while these false positive rates would likely be too high in the context of predicting whole-disk failures and triggering drive replacements, they are acceptable in our context.

Since both drive models report SMART 197 (number of unstable sectors), one might wonder whether the predicted sector reallocations are trivial predictions based on sectors that were already previously known to be unstable. We verify that this is not the case by removing SMART 197 as an input variable, retraining the classifier and still achieving the same results. We also observe no correlation between the SMART 5 and SMART 197, based on correlation coefficients.

When comparing different machine learning methods, we observe that random forests consistently outperform or match the performance of other classifiers. This is encouraging for use in practice as random forests are among the classifiers that are the easiest and fastest to train, as there are very few parameters to tune. The main parameter is the number of trees in the forest, and we find that the results are not very sensitive to this parameter. For example, we find that results are the same for forests with 20, 50, 100 and 200 trees.

On the other hand, some of the other classifiers, in particular neural networks, support vector machines and logistic regression required a considerable amount of tuning as part of a lengthy training process. Despite our extensive efforts in training these models, their performance can only barely and only for small false positive ranges match that of random forests.

Finally, we also repeated training and prediction for two other SMART parameters, S187 and S197. The results are included in the appendix. We see similar trends as for S5, in that random forests match or outperform other predictors and we find that prediction accuracy is high, albeit slightly lower than for S5. We hypothesize that the accuracy for S187 and S197 is slightly lower, because whether a bad sector will affect the counts for these two variables will depend on how the bad sector was discovered (e.g. by a read or a write operation), which is a somewhat random factor that would be hard to predict.

3.2.2 Prediction Results for SSDs

In this section we train classifiers to predict uncorrectable errors for the three types of SSDs that we have data for. Figure 2 shows the results. We observe that, as was the case for HDDs, random forests outperform other classifiers. Sector errors can be predicted with a significant accuracy, albeit somewhat lower accuracy than for HDDs. At a false positive rate of 10% the random forest classifier catches 50-70% of errors. At a false positive rate of 2% the classifier can still catch 50-60% of errors for two of the three models.

We also experimented with training classifiers to predict bad blocks and include the results in the appendix.

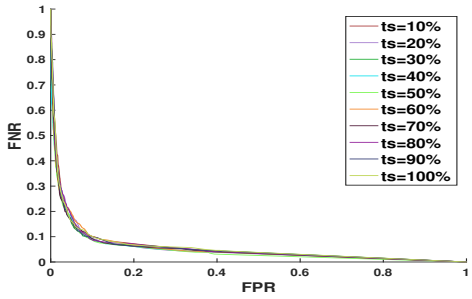


Figure 3: *Training of the random forest classifier for the Hitachi drive with only a fraction of the original training set still leads to nearly identical quality of predictions.*

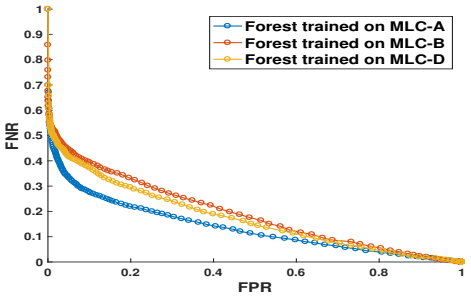
Random forests are again the classifier with the best performance, however prediction accuracy is lower than for uncorrectable errors. A block can be declared due to an uncorrectable read on it, or when (even after a number of retries) write or erase operations fail. Our results seem to indicate that failing write or erase operations are harder to predict than uncorrectable errors.

3.3 Robustness of Predictions

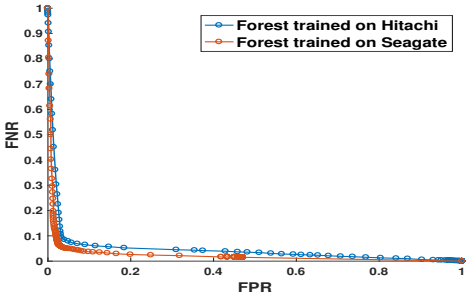
Our predictions in the previous section were based on a relatively large amount of data. One problem in practice is that an operator might not have access to data sets of comparable size (e.g. because the system is smaller or still relatively new, and hence not much data is available yet). To address this problem, we experimented with two different approaches. We repeated the training process, but with significantly smaller amounts of training data. Figure 3 shows the results, when training random forests for the Hitachi HDD on only a fraction of the original training set, ranging from only 10%, to 90% of the training data that was used in the previous section. We observe that prediction quality is hardly affected.

We also experimented with solutions to the problem that no prior data is available when a new type of device is first deployed. In particular, we ran experiments where we use a predictor that was trained on one drive model to make prediction for another drive model. Figure 4 shows the accuracy of predictions when we use random forests trained on MLC-B and MLC-D to predict errors for MLC-A (Figure 4(a)) and we use random forests trained on the Hitachi drive to predict sector errors on the Seagate drive (Figure 4(b)). We observe that while prediction accuracy drops, the quality of the resulting predictions is still high.

This observation is particularly interesting as the drives that were used for model building differed in many important aspects significantly from those we make predictions for. For example, SSD model MLC-B differs from MLC-A in manufacturer and lithography and the



(a) MLC-A



(b) Seagate

Figure 4: *Results for cross-model prediction: We use random forests trained on MLC-B and MLC-D to predict errors for MLC-A in the top figure and we use random forests trained on the Hitachi drive to predict sector errors on the Seagate drive in the bottom figure.*

two drive models have vastly different rates of uncorrectable errors. The two hard disk drives differ in their manufacturer, their capacity and their rate of sector errors. Moreover, the data for the two hard disk drives is based on SMART reporting, which is inconsistent between different manufacturers and models. For example, the Hitachi model does not report all the SMART parameters the Seagate model reports.

4 Tuning Scrub Rates based on Predictions

In the previous section we have developed classifiers to predict future sector errors. In this section, we explore one particular idea for how such predictions could be used to improve the reliability of storage systems.

4.1 Idea

Most storage systems employ a data scrubber to protect against data loss due to sector errors. A scrubber is a background process that periodically performs full-disk scans to proactively detect and correct sector errors. For example, some filesystems, such as ZFS and Btrfs, provide scrubbing at the filesystem level, RAID controllers may initiate periodic scrubs at the block level, and commercial storage systems, such as Netapp's, often support

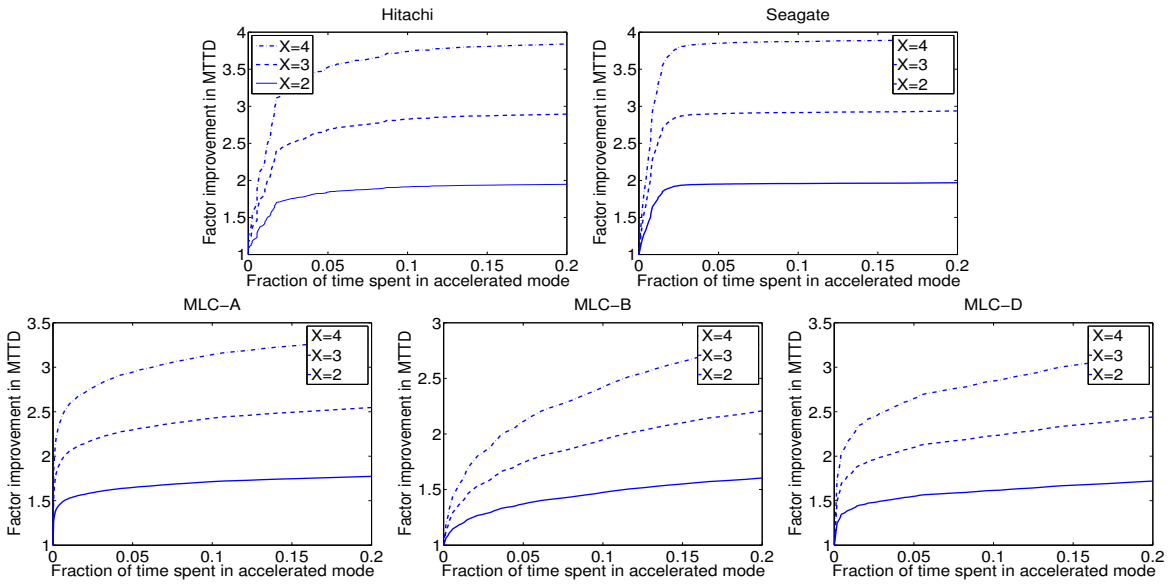


Figure 5: *Simulating the adaptive scrubber. The X-axis shows the fraction of time the scrubber spends in accelerated mode and the Y-axis shows the factor decrease in mean time to error detection. Each line corresponds to a different factor X of acceleration.*

scrubbing at the file and the block level (termed data scrub versus media scrub, respectively [4]).

The goal of the scrubber is to minimize the time between the occurrence of an error and its detection (Mean Time To Detection, MTTD), since during this time the system is vulnerable to data loss (e.g. if a RAID array experiences a whole-disk failure(s) before the sector error is detected and corrected). In addition to minimizing MTTD, a scrubber must ensure that it does not significantly affect the performance of foreground workloads running on the system.

Currently, administrators in practice configure a scrubber to run at a *fixed* scrub rate, which must balance the two goals above: Scrubbing at a fast rate will detect errors more quickly, while a slow scrub speed imposes less load on the system. A common rule of thumb is to complete one full scrub of a drive once a week or bi-weekly.

Instead, we propose to *adapt* the scrub rate dynamically based on the predicted chance of encountering errors, rather than using one fixed scrub rate throughout. This is similar to an idea proposed as future work by Ma et al. [13], who suggest to increase the scrub rate for drives with higher error rates. We use our methods from Section 3.2 to predict errors, and whenever an error is predicted we accelerate the speed at which the system scrubs. When no error is predicted we reduce the scrub speed. Note that errors are rare events, so provided we choose a predictor with a reasonably low rate of false positives, the system should rarely trigger accelerated scrubs.

4.2 Evaluation Setup

We set up a series of simulations to evaluate the effectiveness of an adaptive scrubber compared to a fixed rate scrubber. We consider adaptive scrubbers that switch between two speeds, s_1 and s_2, where s_1 is the slower speed that is used when no errors are predicted and s_2 is the accelerated speed that is used when an error is predicted. One might also consider a continuous spectrum of scrub speeds, but we defer this discussion to Section 4.4.

In all simulations, we rely on predictors based on random forests, as they tended to provide the highest quality predictions. The training of a random forest for a given device type can be configured to achieve different rates of false positives and false negatives. The rate of false positives that is acceptable, will be system dependent based on the system's sensitivity to added workload. For example, for a false positive rate of 2% a system will spend roughly 2 weeks out of the 52 weeks in a year scrubbing at an accelerated rate, without catching any errors. At a lower false positive rate, the system will spend less time in accelerated scrub mode, but will also be slower at catching some errors (as the false negative rate will go up). We therefore experiment with a range of different configurations for the random forest.

We set the s_1 parameter, the default scrub rate when no errors are predicted, to one full disk scrub per week, as this is a common scrub frequency in practice. The second parameter is s_2, i.e. the rate at which the system scrubs in accelerated mode. Again the choice of this parameter will depend on the system's sensitivity to

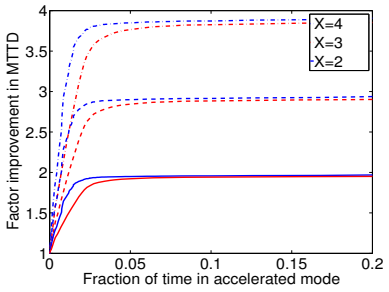


Figure 6: *Results when the adaptive scrubber employs the classifier trained for the Hitachi drive to predict errors on the Seagate drive (red lines) compared to scrub results when prediction for Seagate is done on the models trained on Seagate data (blue lines).*

added workload. If we increase the scrub speed by a factor of X during an accelerated scrub, the MTTD will on average be reduced by a factor of X, but the system experiences a higher load. In our simulations, we therefore experiment with a range of X values. We perform predictions once a week.

4.3 Results

Figure 5 shows the results when simulating the adaptive scrubber using random forests. The X-axis shows the fraction of time the scrubber spends in accelerated mode and the Y-axis shows the factor decrease in mean time to error detection. Each line corresponds to a different factor of acceleration used by the scrubber when an error is predicted. E.g. a 2X factor of acceleration means that the scrub rate is doubled, i.e. if the default scrub rate is to scrub once per week, in accelerated mode the scrubber will complete two scrubs per week.

We observe that even if we limit the time the system is spending in the accelerated scrub mode to 2% of the total time (i.e. we are using predictors with a very low false positive rate), the factor decrease in mean time to error detection is significant. For example, in the case of the hard disk drives when the scrub speed is doubled upon an error prediction we detect errors on average 1.7-1.8X faster than a fixed rate scrubber. Even for the SSDs, for which the classifiers' predictions were less accurate, the savings are still significant. E.g. for MLC-A and MLC-D errors are detected on average 1.4-1.5X faster when the scrub speed is doubled upon an error prediction, even if the total time spent in accelerated mode is limited to 2%.

We also consider the case where predictors for a given drive model are not available, e.g. because the drive model is new and has just been deployed and where predictions are obtained by using classifiers trained for a different drive model. Recall that in Section 3.3 we predicted errors for the Seagate model based on a classifier trained for the Hitachi model. Figure 6 shows the results when the adaptive scrubber adjusts scrub speeds for the Seagate drive based on predictions from the classifier trained on the Hitachi drive. We observe that the improvements in the mean time to detection are still very good.

4.4 Refinements and Practical Concerns

While our results provide a proof-of-concept for prediction-based tuning of scrub rates, the methods could be further refined. For example, we only considered a scrubber that alternates between two speeds. One might further improve results by adjusting scrub speeds on a continuous spectrum, based on the certainty of an error prediction. In their raw form forests (and most of the other methods we considered) produce error probabilities, rather than a binary error versus no-error prediction. The binary predictions are produced by applying some thresholds to the produced probabilities. One could instead use the raw probabilities and adjust the scrub speed along a continuous spectrum based on how large the current error probability is, and might do better than in the simple two-speed scrubber.

One practical concern with accelerating scrub speeds is the impact that the additional load has on the system. There are tools available to an administrator to mitigate possible negative effects on foreground workload. Besides the obvious one of limiting the increase in scrub speed to some maximum value the administrator deems tolerable for their system, an administrator could also put a limit on the maximum amount of time the system spends in accelerated scrubs. Moreover, there are techniques that have been suggested recently [1] to reduce the impact of storage maintenance workloads on the system that can be applied to scrubbing as well.

A secondary concern might be that the additional load imposed on the system might induce new errors. While solid state drives are known to wear out faster under heavy write workloads, scrub operations consist of reads only, and two recent papers [15, 22] independently show that there is no correlation between the number of reads and the number uncorrectable errors in a system. Similarly, for hard disk drives one might expect writes to have a correlation with sector errors, as an incorrect write (e.g. a high-fly write) might be the cause of sector errors, but a study of field data [21] finds no correlation between read operations and sector errors.

5 Other Use Cases for Error Prediction

This section proposes and discusses the use of error predictors as adaptive policy-enforcing mechanisms in storage systems. We leave the detailed exploration of these use cases to future work.

5.1 Tuning Drive Internal Mechanisms

5.1.1 Adaptive Error Correcting Codes

There has been some recent work in the flash community to equip drives with adaptive error correcting codes [7, 10]. The original motivation was that the reliability of flash cells changes over their lifetime, so one could use a smaller, less powerful code at the beginning of a drive's life and switch to larger, but stronger codes later on, as the drive ages. Rather than using the age of a drive as an indicator whether the drive should switch to the stronger ECC, it would be natural to use an error predictor, and trigger the switch when predictions indicate that the drive is more likely to develop uncorrectable errors.

5.1.2 Proactive Retirement of Blocks or Chips

Our predictors only predict whether a drive will experience sector errors or bad blocks, or not. We do not predict the location of any future errors (i.e. which block and which chip), because the field data available to us does not include location information. However, it is likely that errors occur at or near those locations that are responsible for early symptoms of errors (e.g. various types of prior errors).

It would be interesting to also predict the precise location of future errors, and investigate whether these predictions can be used to proactively retire a block or a chip. We hope this work will encourage storage manufacturers and others to explore making such predictions using data available inside drives, and exposing this information to consumers.

5.2 Tuning the Cache Policy in SSD Caches

SSDs are not only used for persistent storage of data, but also as a caching layer. In the case of write-through caches, errors do not pose a risk for data loss. They just translate to higher read latency for data affected by errors, as accessing it will turn into a cache miss. In the case of write-back caches, errors create the potential for data loss if they affected dirty data in the cache that has not yet been flushed to persistent storage. One could therefore consider a policy where an error predictor is used to switch the cache policy from write-back to write-through when a predictor indicates future errors.

5.3 Tuning Filesystem Mechanisms

Many filesystems contain mechanisms to protect against sector errors. These range from replicating important data structures (such as a filesystem's super-block), to adding checksums for metadata (such as inodes), replicating data as well as metadata, or even RAID-5 or RAID-6 level data protection in the case of ZFS. An interesting area of future work is to explore which of these mechanisms could be dynamically enabled using error predictors. This would be particularly useful if filesystem mechanisms such as metadata replication had some advance warning about which specific blocks or chips were likely to fail (as discussed in Section 5.1.2).

6 Related Work

6.1 Predicting Errors in Storage Systems

To the best of our knowledge there is no prior work on predicting *partial* drive failures, such as sector errors on hard disks and uncorrectable errors and bad blocks in solid state drives. Instead prior work on drive reliability predictions focuses on predicting *complete* drive failure for HDDs [8, 9, 11–13, 16, 24] and SSDs [17].

The motivations and requirements for predicting complete drive failure are very different from our work on predicting partial drive failures. The goal is to use predictions of drive failures in order to initiate proactive drive replacement before the failure occurs. Such predictions require an extremely low false positive rate, since the unnecessary replacement of healthy drives is associated with significant costs. In contrast, we are interested in predicting partial drive failures to guide the system to take lighter-weight proactive measures, such as increasing the scrub rate. Another difference between our work and much of the prior work is that training and test instances in prior work were often not drives deployed in productions systems, but rather drives run in a controlled environment by the manufacturer [11,16], or the data sets that were used were of very limited size (e.g. less than 20 faulty drives in [8, 9]).

Finally, our work explores a wide range of modern machine learning techniques, which subsume most of the techniques used in prior work. Hamerly et al. [9] use Bayesian approaches. We applied logistic regression instead, as naive Bayes methods make a strong assumption on the conditional independence of the input variables. Hughes et al. [11] use statistical hypothesis tests, but find in later work that SVMs perform better [16]. Our work includes SVMs, as well as a number of other techniques not explored by [11,16]. The work by Pinheiro et al. [20] does not attempt to predict drive failures, however they observe a correlation between some SMART parameters and drive failures. The work by Ma et al. [13] uses a simple threshold-based prediction. They find that by putting a threshold on the number of sector reallocations drive failure can be predicted well. We experimented with this approach and find that it performs poorly on the hard disk data, compared to the machine learning techniques, and is limited to results with very high false negative rates for the solid state drives.[1]

[1] For space reasons we do not include detailed results for thresholding in this work, but instead refer the reader to a tech-report [14].

We are aware of only three papers that use machine-learning based techniques on large-scale field data and again they are predicting whole drive failures [12,17,24]. Two of the papers focus on HDDs and find that CART models outperform SVMs and neural networks [12, 24]. Narayanan et al. [17] show that random forests can predict fail-stop events in SSDs (events that lead to server shut-down and typically drive replacement), albeit accuracy is lower than that reported in papers predicting HDD failure. Our work includes all these methods, among others, and we find that forests are superior to the other methods for predicting sector errors.

6.2 Improving Scrubbers

Prior work on scrubbing is mostly focused on minimizing the impact of scrub operations on foreground traffic, e.g. by trying to submit scrub operations during idle times [2] or by piggy-backing it on workload operations [1], or reducing the time to detect errors by modifying the order in which sectors are scrubbed [18]. The only work we are aware off that adjusts scrub speeds is by Paris et al. [19] and proposes to perform an expedited scrub run after a whole-disk failure in a RAID-6 array. Our idea of adjusting scrub speed based on error predictions is similar to Ma's [13] future work suggestion of increasing scrub rates for drives with high error rates. Our work on adjusting scrub speed based on error predictions is orthogonal to work on minimizing the impact of scrub operations or the optimal ordering of scrub operations within a scrub and can be used in conjunction with any of these techniques.

7 Sharing of Data

All the data on hard disk drives used in the paper is already publicly available on the Backblaze home page, so all our results are reproducible by other researchers. The data on solid state drives has been shared with us by Google, and we are currently working with our collaborators at Google towards sharing this data publicly.

8 Conclusion

This paper explores the use of machine learning to make storage systems more reliable in the face of latent sector errors. We experiment with a wide variety of machine learning techniques and find that sector errors in hard disk drives and solid state drives can be predicted accurately with classifiers based on random forests, which are easy to train and to parameterize. We show that training is robust even for small training sets or when training data comes from a different drive model than the target system. We also discuss a number of possible use cases for improving storage system reliability through the use of sector error predictors. We evaluate one such use case in detail: We show that the mean time to detecting errors (and hence the window of vulnerability to data loss) can be greatly reduced by adapting the speed of a scrubber based on error predictions.

Appendix

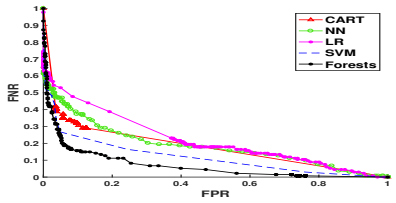


Figure 7: *Predicting SMART 197 for Hitachi*

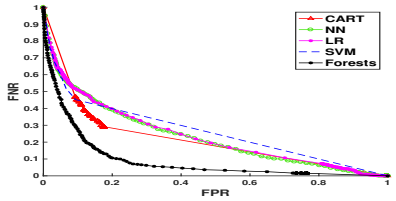


Figure 8: *Predicting SMART 187 for Seagate*

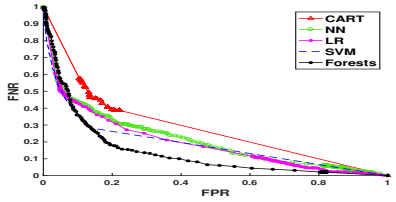


Figure 9: *Predicting SMART 197 for Seagate*

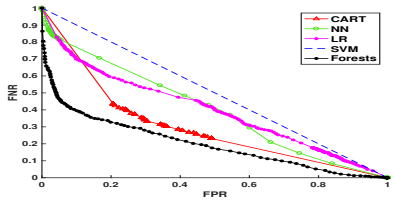


Figure 10: *Predicting bad blocks for MLC-A*

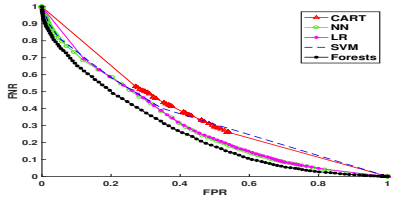


Figure 11: *Predicting bad blocks for MLC-B*

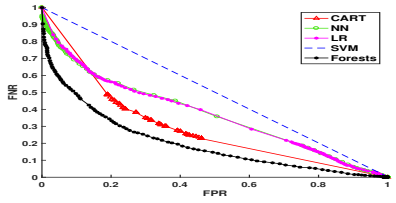


Figure 12: *Predicting bad blocks for MLC-D*

9 Acknowledgments

We would like to thank the Usenix ATC '17 anonymous reviewers, and our shepherd Fred Douglis for the valuable feedback and the many good questions they brought up. While we did not have room to address all of them in this paper, we refer the reader to an extended version of our paper, including additional experiments, which addresses many of these questions [14]. We would also like to thank Arif Merchant for his continued efforts to make the Google SSD data available.

References

[1] AMVROSIADIS, G., BROWN, A. D., AND GOEL, A. Opportunistic storage maintenance. In *Proceedings of the 25th Symposium on Operating Systems Principles* (New York, NY, USA, 2015), SOSP '15, ACM, pp. 457–473.

[2] AMVROSIADIS, G., OPREA, A., AND SCHROEDER, B. Practical scrubbing: Getting to the bad sector at the right time. In *Dependable Systems and Networks (DSN), 2012 42nd Annual IEEE/IFIP International Conference on* (june 2012), pp. 1 – 12.

[3] BACKBLAZE. The backblaze hard drive data and stats. https://www.backblaze.com/b2/hard-drive-test-data.html, 2016.

[4] BAIRAVASUNDARAM, L. N., GOODSON, G. R., PASUPATHY, S., AND SCHINDLER, J. An analysis of latent sector errors in disk drives. In *Proceedings of the 2007 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems* (New York, NY, USA, 2007), SIGMETRICS '07, ACM, pp. 289–300.

[5] BREWER, E., YING, L., GREENFIELD, L., CYPHER, R., AND TS'O, T. FAST'16 Keynote talk: Disks for Data Centers. Tech. rep., Google, 2016.

[6] CHANG, C.-C., AND LIN, C.-J. The libsvm library for matlab. https://www.csie.ntu.edu.tw/ cjlin/libsvm, 2017.

[7] CHEN, T.-H., HSIAO, Y.-Y., HSING, Y.-T., AND WU, C.-W. An adaptive-rate error correction scheme for nand flash memory. In *VLSI Test Symposium, 2009. VTS'09. 27th IEEE* (2009), IEEE, pp. 53–58.

[8] GOLDSZMIDT, M. Finding soon-to-fail disks in a haystack. In *Proceedings of the 4th USENIX Conference on Hot Topics in Storage and File Systems* (Berkeley, CA, USA, 2012), HotStorage'12, USENIX Association, pp. 8–8.

[9] HAMERLY, G., ELKAN, C., ET AL. Bayesian approaches to failure prediction for disk drives. In *ICML* (2001), vol. 1, Citeseer, pp. 202–209.

[10] HARATSCH, E. F. Nand flash media management algorithms. In *Flash Memory Summit* (2016).

[11] HUGHES, G. F., MURRAY, J. F., KREUTZ-DELGADO, K., AND ELKAN, C. Improved disk-drive failure warnings. *IEEE Transactions on Reliability 51*, 3 (2002), 350–357.

[12] LI, J., JI, X., JIA, Y., ZHU, B., WANG, G., LI, Z., AND LIU, X. Hard drive failure prediction using classification and regression trees. In *44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2014, Atlanta, GA, USA, June 23-26, 2014* (2014), pp. 383–394.

[13] MA, A., TRAYLOR, R., DOUGLIS, F., CHAMNESS, M., LU, G., SAWYER, D., CHANDRA, S., AND HSU, W. Raidshield: Characterizing, monitoring, and proactively protecting against disk failures. *Trans. Storage 11*, 4 (Nov. 2015), 17:1–17:28.

[14] MAHDISOLTANI, F., STEFANOVICI, I., AND SCHROEDER, B. Improving storage system reliability with proactive error prediction. Tech. Rep. 633, University of Toronto, 2017.

[15] MEZA, J., WU, Q., KUMAR, S., AND MUTLU, O. 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* (New York, NY, USA, 2015), SIGMETRICS '15, ACM, pp. 177–190.

[16] MURRAY, J., HUGHES, G., AND KREUTZ-DELGADO, K. Machine learning methods for predicting failures in hard drives. *Journal of Machine Learning Research* (2005).

[17] NARAYANAN, I., WANG, D., JEON, M., SHARMA, B., CAULFIELD, L., SIVASUBRAMANIAM, A., CUTLER, B., LIU, J., KHESSIB, B., AND VAID, K. Ssd failures in datacenters: What? when? and why? In *Proceedings of the 9th ACM International on Systems and Storage Conference* (2016), SYSTOR '16, ACM, pp. 7:1–7:11.

[18] OPREA, A., AND JUELS, A. A clean-slate look at disk scrubbing. In *Proceedings of the 8th USENIX Conference on File and Storage Technologies* (Berkeley, CA, USA, 2010), FAST'10, USENIX Association, pp. 5–5.

[19] PARIS, J.-F., SCHWARZ, T., AMER, A., AND LONG, D. D. E. Improving disk array reliability through expedited scrubbing. In *Proceedings of the 2010 IEEE Fifth International Conference on Networking, Architecture, and Storage* (Washington, DC, USA, 2010), NAS '10, IEEE Computer Society, pp. 119–125.

[20] PINHEIRO, E., WEBER, W.-D., AND BARROSO, L. A. Failure trends in a large disk drive population. In *Proceedings of the 5th USENIX Conference on File and Storage Technologies* (2007), FAST '07, USENIX Association.

[21] SCHROEDER, B., DAMOURAS, S., AND GILL, P. Understanding latent sector errors and how to protect against them. In *Proceedings of the 8th USENIX Conference on File and Storage Technologies* (Berkeley, CA, USA, 2010), FAST'10, USENIX Association.

[22] SCHROEDER, B., LAGISETTY, R., AND MERCHANT, A. Flash reliability in production: The expected and the unexpected. In *Proceedings of the 14th Usenix Conference on File and Storage Technologies* (Berkeley, CA, USA, 2016), FAST'16, USENIX Association, pp. 67–80.

[23] WIKIPEDIA. S.m.a.r.t. https://en.wikipedia.org/wiki/S.M.A.R.T.

[24] ZHU, B., WANG, G., LIU, X., HU, D., LIN, S., AND MA, J. Proactive drive failure prediction for large scale storage systems. In *IEEE 29th Symposium on Mass Storage Systems and Technologies, MSST 2013, May 6-10, 2013, Long Beach, CA, USA* (2013), pp. 1–5.