Proactive Drive Failure Prediction for Large Scale Storage Systems

Bingpeng Zhu[1], Gang Wang*[1], Xiaoguang Liu*[2], Dianming Hu[3], Sheng Lin[4], Jingwei Ma*[1]
[1] *Nankai-Baidu Joint Lab, College of Information Technical Science, Nankai University, Tianjin, China*
[2] *College of Software, Nankai University, Tianjin, China*
[3] *Baidu Inc., Beijing, China*
[4] *School of Computer and Communication of Engineering, Tianjin University of Technology, Tianjin, China*
{nkuzbp, wgzwp, liuxg74}@hotmail.com, hudianming@baidu.com, {shshsh.0510, mjwtom}@gmail.com

Abstract—**Most of the modern hard disk drives support Self-Monitoring, Analysis and Reporting Technology (SMART), which can monitor internal attributes of individual drives and predict impending drive failures by a thresholding method. As the prediction performance of the thresholding algorithm is disappointing, some researchers explored various statistical and machine learning methods for predicting drive failures based on SMART attributes. However, the failure detection rates of these methods are only up to $50\% \sim 60\%$ with low false alarm rates (FARs). We explore the ability of Backpropagation (BP) neural network model to predict drive failures based on SMART attributes. We also develop an improved Support Vector Machine (SVM) model. A real-world dataset concerning $23,395$ drives is used to verify these models. Experimental results show that the prediction accuracy of both models is far higher than previous works. Although the SVM model achieves the lowest FAR (0.03%), the BP neural network model is considerably better in failure detection rate which is up to 95% while keeping a reasonable low FAR.**

I. Introduction

Nowadays, large scale storage systems usually deploy massive hard disk drives as primary data storage device. To provide high reliability in such systems, reactive fault-tolerant techniques, such as replication and erasure code are often used. Currently, almost all hard drive manufacturers have implemented Self-Monitoring, Analysis and Reporting Technology (SMART) [1] in their products, which monitors internal attributes of individual drives and raises an alarm if any attribute exceeds its threshold. However, it has been estimated that the thresholding algorithm can only reach a failure detection rate of $3-10\%$ at 0.1% false alarm rate (FAR) [8]. Some statistical and machine learning methods have been proposed to build better prediction models based on the SMART attributes [5], [6], [7], [8], [13]. However, their failure detection rates are only up to $50\% \sim 60\%$ with low FARs.

In this paper, we explore building drive failure prediction model based on Backpropagation (BP) neural network [10]. An improved Support Vector Machine (SVM) [12] model is also proposed. We use new training and detection strategies to improve the prediction accuracy. Both algorithms are trained and tested on a real-world dataset concerning $23,395$ drives, and perform much higher prediction accuracy than all of the previous works [5], [6], [7], [8], [13]. We show that such high prediction accuracy can significantly improve the reliability of storage systems.

II. Related work

To improve failure prediction accuracy of SMART, Hamerly and Elkan [5] employed two Bayesian approaches (NBEM and naive Bayes classifier) to build prediction models. Both methods were tested on a dataset concerning $1,936$ drives. They achieved failure detection rates of $35-40\%$ for NBEM and 55% for naive Bayes classifier at about 1% FAR.

Another study on drive failure prediction was performed by Hughes *et al.* [6]. They used Wilcoxon rank-sum test to build prediction models. They proposed two different strategies: multivariate test and ORing single attribute test. Their methods were tested on $3,744$ drives. The highest detection rate achievable was 60% with 0.5% FAR.

Murray *et al.* [7] compared the performance of SVM, unsupervised clustering, rank-sum test and reverse arrangements test. In their subsequent work [8], they developed a new algorithm termed multiple-instance naive Bayes (mi-NB). They found that, on the dataset concerning 369 drives, rank-sum test outperformed SVM for certain small set of SMART attributes (28.1% failure detection at 0% FAR). When using all features, SVM achieved the best performance of 50.6% detection with 0% FAR.

III. Modeling methodology

A. Support vector machine

SVM [12] is a supervised machine learning method for classification and regression. Given a set of training samples from two classes, SVM algorithm can find the best decision hyperplane that separates the two classes. For those datasets that are non-linearly separable, SVM algorithm can implicitly map the training data into a higher dimensional feature space by a kernel function. In this higher dimensional space, the two classes can be separated linearly.

Compared with previous SVM models for drive failure prediction [7], [8], our SVM model uses both SMART attributes and their change rates as features. Besides, samples from different time windows are used to train different models.

978-1-4799-0218-7/13/$31.00 ©2013 IEEE

B. *Artificial neural network*

Artificial neural networks (ANN) [10] can be viewed as functions that convert a vector of input variables to another vector of output variables. A typical method for training ANN is the BP algorithm [10]. Figure 1 shows the architecture of the neural network for drive failure prediction. The nodes are called artificial neurons and arranged in three layers: input layer, hidden layer, and output layer. The neurons in adjacent layers are connected with different weights.

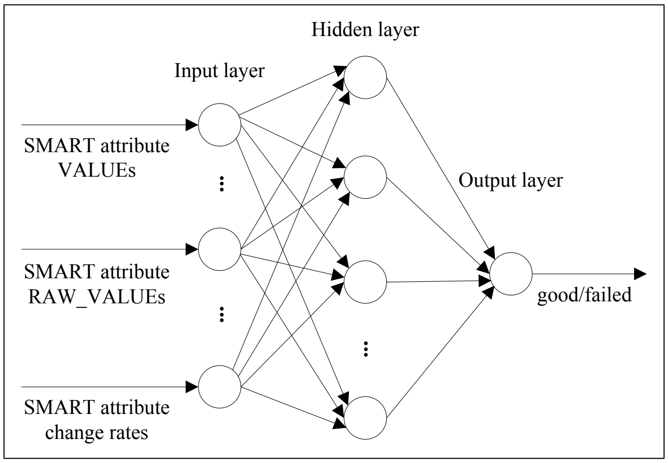

Fig. 1. Three-layer BP neural network.

The BP learning algorithm has two phases: a feed-forward stage and a back-propagation stage. In the feed-forward stage, each neuron calculates weighted sum of input neurons' values and then applies an activation function to the sum as the output of this node. The values flow until they reach the output layer. In the back-propagation stage, the weights between neurons are updated by the learning rule for reducing the discrepancies between actual output and the target value.

IV. Dataset description and preprocessing

Our SMART dataset was collected from a single running datacenter of Baidu Inc. with uniform environment.[1] There are $23,395$ drives in the dataset and all of them are with the same model. SMART attribute values were sampled from each working drive at every hour. Each drive is labeled *good* or *failed*, with only 433 drives in the failed class and the rest ($22,962$ drives) in the good class. For good drives, the samples in a week are kept in the dataset. For failed drives, samples in a longer time period (20 days before actual failure) are saved. This dataset is at least an order of magnitude larger than those used in previous studies [5], [6], [7], [8], [13].

A. *Feature selection and construction*

We can read out 23 meaningful attribute values from every drive at a time, but some attributes are useless for failure prediction since they keep unchanged during operation. We get rid of these attributes. Table I lists the remaining 10 useful SMART attributes for build prediction models.

TABLE I
Selected SMART attributes as features.

ID #	Attribute Name
1	Raw Read Error Rate
3	Spin Up Time
5	Reallocated Sectors Count
7	Seek Error Rate
9	Power On Hours
187	Reported Uncorrectable Errors
189	High Fly Writes
194	Temperature Celsius
195	Hardware ECC Recovered
197	Current Pending Sector Count

Each SMART attribute has a six-byte width raw value (RAW_VALUE) and a normalized value (VALUE) ranging from 1 to 253 [1]. The format of RAW_VALUE is vendor-specific. However, there are still two interpretable RAW_VALUEs in our dataset (i.e. those of attributes #5 and #197 in Table I). Since RAW_VALUE is more sensitive to the drive's health status, we also select the two RAW_VALUEs for building models.

We observed that for some (but not all) failed drives, several particular attributes show significant downward trend over time. While for almost all of the healthy drives, these attributes keep unchanged or change little. They are VALUEs of attributes #1, #5, #187, #195, #197 and RAW_VALUEs of attributes #5 and #197. For every sample, we calculate the absolute differences between the current values of these attributes and their corresponding values six hours ago as features. At last, each sample in our dataset has 19 features including 10 VALUEs, 2 RAW_VALUEs, and 7 change rates.

B. *Feature normalization*

Data normalization can promise a fair comparison between different feature values in machine learning algorithms. The formula of data normalization we used is given below:

$$x_{normal} = 2 \times \frac{x - x_{min}}{x_{max} - x_{min}} - 1 \qquad (1)$$

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 in our dataset. When applying the failure prediction model to a real storage system, we should use the maximum and minimum values in the existing dataset.

V. Experimental results and discussion

We divide all the drives randomly into training and test sets. The training set consists of 70% of all the good and failed drives, and the remaining 30% of the drives are in the test set. For each good drive in the training set, we randomly choose 4 samples as "good" samples (hereinafter referred as *negative samples*) to train models. We choose 4 samples per good drive because it can eliminate the bias of a single drive's sample in a particular hour and provide enough information to describe the good status of the drive. For failed drives in the training set, we choose samples collected within a certain time window before the actual drive failure as "failed" samples

[1] The dataset is now available at `http://pan.baidu.com/share/link?shareid=189977&uk=4278294944`.

(hereinafter referred as *positive samples*) to train models. To determine the time window resulting in the best prediction performance, samples in the last available 12 hours, 24 hours, 2 days, and 4 days are chosen as positive samples to train models, respectively. When we test the models, we check the samples in the test set sequentially for each drive, and predict that the drive is going to fail if any of its samples is classified as failed. Otherwise, it is classified as a good drive.

The detection rate is defined as the fraction of failed drives that are predicted correctly as failed. FAR means the fraction of good drives that are mis-classified as failed. Since good drives are the absolute majority in reality, a high FAR implies too many false-alarmed drives and results in heavy processing cost. We are concerned with keeping a low FAR and we only present the parameter and strategy combinations achieving FARs lower than 5%.

A. *SVM results*

LIBSVM [2] is used to implement the SVM model. We label good drive samples with $+1$ and failed drive samples with -1, respectively. Parameters of LIBSVM are set as follows: *svm-type* = C-SVC, *kernel-type* = radial basis function, *cost C* = 10. The penalty parameters w for classes $+1$ and -1 (denoted by w_{+1} and w_{-1} respectively) can be adjusted to trade off between the failure detection rate and FAR. Other parameters are set to default values. Positive samples from the four time windows mentioned above are used to train different SVM models while keeping negative training samples fixed.

Figure 2 shows the failure prediction performance of our SVM models in the form of Receiver Operating Characteristic (ROC) curve. Each point in Figure 2 denotes the result of a particular time window with a particular penalty parameter pair (w_{+1}, w_{-1}). The points are not linked to curves for clarity. It shows that as the time window becomes earlier, the detection rate increases at the expense of an increasing FAR. The reason is that an earlier time window introduces more positive samples closer to negative samples, which is in favor of the detection rate and against FAR. Since we want a low FAR, we had better use a short time window to train models, such as the last available 12 or 24 hours before the drive failure. Figure 2 shows that our SVM model gets a failure detection rate of 68.5% with the lowest FAR of 0.03% when using the 12 hours time window and the penalty parameter pair $(5,1)$. When the penalty parameter pair is adjusted to $(1,2)$, the SVM model gets a detection rate of 80.0% with 0.3% FAR. This result is far higher than the prediction accuracy achieved by pervious SVM models [7], [8].

Another important measurement is how long in advance we can detect an impending drive failure. For the point of 68.5% detection rate and 0.03% FAR in Figure 2, the distribution of lead time of correct predictions is shown in Figure 3. It has an average lead time of 334 hours, which is long enough for users to take actions before the drive failure actually occurs. For other points in Figure 2, their average lead times are about $330 \sim 360$ hours. The results are much better than the average lead time of about 4 days reported in [8].

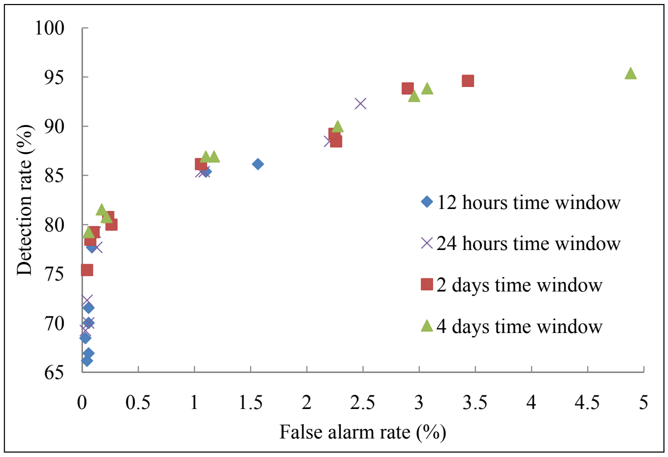

Fig. 2. Failure prediction performance of SVM models.

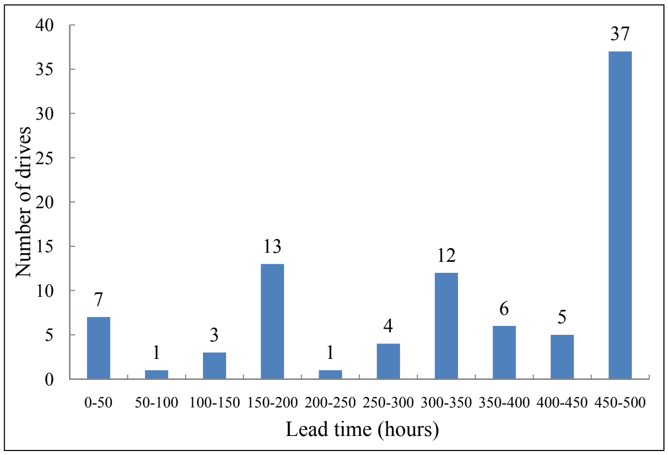

Fig. 3. Distribution of lead time of SVM model.

To verify the effectiveness of the seven features reflecting change rates, we also train SVM models using samples without change rate features, and compare them with the models trained with change rates. The result is illustrated in Figure 4. The time window is set to 12 hours. In general, the ROC curve of the SVM models trained with change rates shows higher detection rate and lower FAR than that of the models trained without change rates. Although the performance gap is narrow, the $3 \sim 4$ percent improvement in detection rate will bring more than 10 percent improvement in MTTDL (Mean Time To Data Loss) because the MTTDL of storage system increases super linearly as the prediction accuracy improves [3].

B. *BP neural network results*

We implement the BP neural network shown in Figure 1. The numbers of nodes in the three layers are 19, 30, and 1, respectively. The target values for the node in output layer are 0.9 and 0.1 for healthy and failed drives, respectively. Both hidden and output layers use sigmoid function as activation function. We set the maximum number of iterations to 400 and the learning rate to 0.1.

Table II shows prediction accuracy and average lead time of BP network models trained by samples in different time windows. It shows the same trend as the SVM results, the earlier time window, the higher the detection rate and FAR.

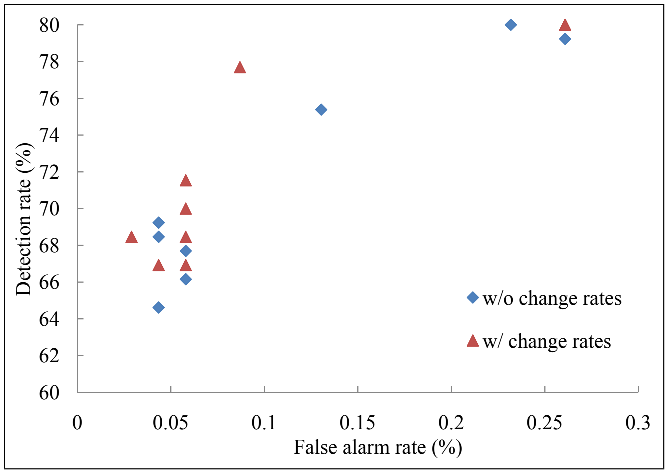

Fig. 4. Effectiveness of change rate features.

The distribution of lead time for BP network model is similar to Figure 3. The prediction performance of BP network models is much better than those reported by previous works [5], [6], [7], [8], [13] and those of our improved SVM models.

TABLE II
PREDICTION RESULTS OF BP NETWORK MODELS.

Time window	FAR(%)	Detection rate(%)	Lead time (hours)
12 hours	0.48	94.62	360.4
24 hours	1.14	97.69	355.9
2 days	1.39	99.23	357.0
4 days	2.26	100.0	356.8

Although BP network models achieve far higher failure detection rate than SVM models, they tend to achieve slightly higher FAR. The lowest FAR achieved by BP network model is 0.48%, which is worse than the best FAR (0.03%) of the SVM model. To reduce the FAR of BP network models, we propose a voting-based failure detection algorithm. Instead of making a prediction by a single sample, this algorithm uses the last N consecutive samples at every time point for prediction: if more than $N/2$ samples vote for failed, the drive is classified as failed. This approach is shown in Algorithm 1. Note that the ordinary non-voting failure detection method used in previous experiments can be viewed as a special case when $N = 1$. Figure 5 shows the prediction results of BP neural network models when using the voting-based detection algorithm. The points on each curve are achieved by setting $N = 1$, 3, 5, 7, 9, 11, and 13 from right to left. When we select a larger N, the FAR of the BP network models decreases while the detection rate is changeless or slightly decreases. That is, the voting-based algorithm effectively reduces the FAR of BP network models while keeping a reasonable detection rate. When applying this approach to SVM models, it shows the same effect. However, since the best FAR of 0.03% achieved by the SVM model is already quite low, this approach can not lower it further.

Algorithm 1 Voting-based failure detection algorithm

```
Input: The sample set $S[1..t]$ of the drive, the BP prediction
    model $BP()$ which returns 0 if the input sample is classi-
    fied as good and 1 otherwise, and the voter turnout $N$
Output: good or failed
Begin
$C[1..N] = 0$
for $i = 1$ to $t$ do
    $C[((i-1) \mod N)+1] = BP(S[i])$
    if $\sum_{j=1}^{N} C[j] > N/2$ then
        return failed
    end if
end for
return good
End
```

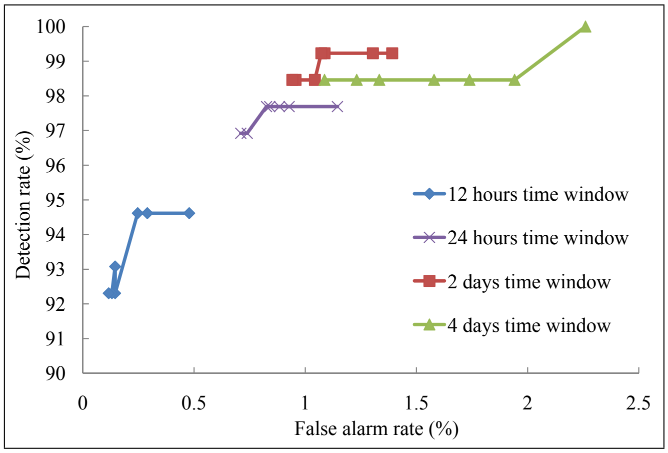

Fig. 5. Failure prediction performance of BP network models using the voting-based detection method.

We also apply a boosting method (AdaBoost [9]) to improve the prediction performance of BP neural network models. When setting the number of weak classifiers to 10, prediction results of AdaBoost-enhanced BP network models using the voting-based detection method are shown in Figure 6. Compared to the results of plain BP network models in Figure 5, AdaBoost algorithm indeed improve the prediction performance. However, the performance gap is narrow because the plain BP network models have already achieved very good prediction accuracy. Generally, by combining AdaBoost and the voting-based detection method, we can improve prediction performance in several ways: higher detection rate, lower FAR, and sometimes both.

In a word, the experimental results show the big advantage of the BP neural network model in prediction accuracy over the SVM model. Considering results reported in pervious literatures [5], [6], [7], [8], [13], we can say that our BP network model obtains the best prediction accuracy by far.

C. *benefit of failure prediction*

Eckart *et al.* [3] devised a Markov model to analyze $N+1$ RAID systems (i.e. systems using parity) with drive failure prediction. We use this model to compute the MTTDLs of such systems with different prediction models. The MTTDLs of $N+1$ and $N+2$ RAID systems without failure prediction are also computed using formulas in [4] for comparison. The same MTTF (Mean Time To Failure) and MTTR (Mean Time

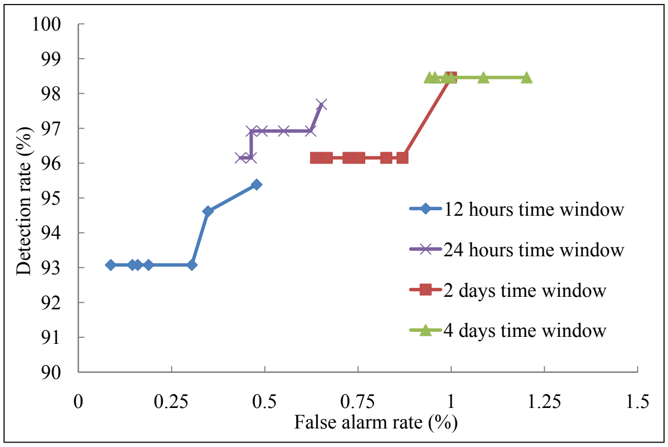

Fig. 6. Failure prediction performance of AdaBoost-enhanced BP network models using the voting-based detection method.

To Repair) of a single hard drive as those in [3] are used. For our BP neural network and SVM models, the result of 94.62% detection rate with 360 hours lead time and the result of 68.5% detection rate with 334 hours lead time are used, respectively. Figure 7 shows the MTTDL for each configuration with different sizes up to 2000 drives. Although such a large RAID is impossible in real world large distributed file systems such as a Hadoop file system [11], it can be used to estimate the reliability of this kind of systems. This kind of systems generally store three replicas by default for each data chunk, and spread chunk replicas across machines and racks. So each drive may share chunks with drives on different machines and racks. Therefore, such a system with a replication factor three (two) has a MTTDL between the MTTDL of an $N+2$ ($N+1$) RAID and that of a three-way (two-way) mirroring system. This estimation may be too coarse, however, we are interested in showing the improvement of MTTDL.

Figure 7 shows that the BP neural network and SVM models improve the MTTDL by several orders of magnitude. An $N+1$ RAID using BP network prediction model achieves the same level of MTTDL as an $N+2$ RAID without prediction, and exceeds the latter when the system size approaches 1000. This result suggests that, using our failure prediction models, we can replace complex fault-tolerant mechanisms with the simpler ones to lower the storage cost and reduce read/write overhead while maintaining the same reliability level.

VI. Conclusions and Future work

In this paper, we explore the ability of BP neural network to predict drive failures based on SMART attributes. We also build an improved SVM model. Besides, new training and detection strategies are proposed to improve the prediction performance. Compared with previous studies, our prediction models achieve much higher prediction accuracy. The SVM model achieves the lowest FAR (0.03%), and the BP neural network model is far superior in detection rate which is more than 95% while keeping a reasonable low FAR. We believe our models are extremely promising for future proactive failure prediction systems.

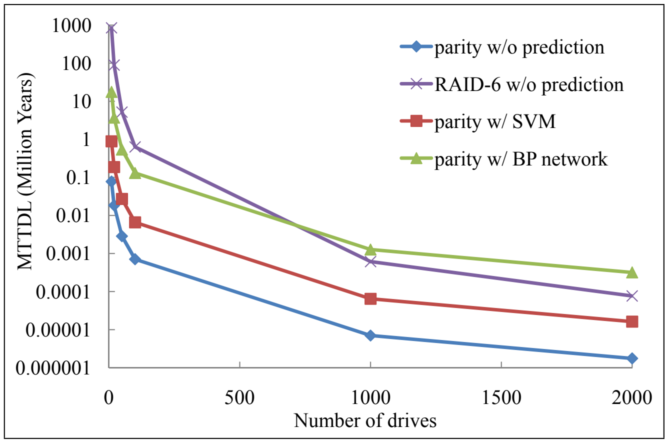

Fig. 7. MTTDL of RAID with varying sizes.

There are several problems to be solved in the future. The prediction models need to be further verified in real storage systems. Besides, automatic data migration for the storage system after failure prediction is an important research topic.

Acknowledgment

This paper is partially supported by NSFC of China (60903028, 61070014, 61170301), Key Projects in the Tianjin Science & Technology Pillar Program (11ZCKFGX01100).

References

[1] B. Allen, “Monitoring hard disks with SMART,” *Linux Journal*, no. 117, Jan 2004.

[2] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1–27:27, Apr 2011.

[3] B. Eckart, X. Chen, X. He, and S. L. Scott, “Failure prediction models for proactive fault tolerance within storage systems,” in *IEEE International Symposium on Modeling, Analysis and Simulation of Computers and Telecommunication Systems*, Sep 2008, pp. 1–8.

[4] G. A. Gibson and D. A. Patterson, “Designing disk arrays for high data reliability,” *Journal of Parallel and Distributed Computing*, vol. 17, no. 1-2, pp. 4–27, Jan 1993.

[5] G. Hamerly and C. Elkan, “Bayesian approaches to failure prediction for disk drives,” in *Proceedings of the 18th International Conference on Machine Learning*, June 2001, pp. 202–209.

[6] G. F. Hughes, J. F. Murray, K. Kreutz-Delgado, and C. Elkan, “Improved disk-drive failure warnings,” *IEEE Transactions on Reliability*, vol. 51, no. 3, pp. 350–357, Sep 2002.

[7] J. F. Murray, G. F. Hughes, and K. Kreutz-Delgado, “Hard drive failure prediction using non-parametric statistical methods,” in *Proceedings of the International Conference on Artificial Neural Networks*, June 2003.

[8] ——, “Machine learning methods for predicting failures in hard drives: A multiple-instance application,” *Journal of Machine Learning Research*, vol. 6, pp. 783–816, May 2005.

[9] R. Rojas, “AdaBoost and the super bowl of classifiers a tutorial introduction to adaptive boosting,” Freie University, Berlin, Tech. Rep., Dec 2009.

[10] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach (2nd Edition)*. Prentice Hall, 2003.

[11] K. Shvachko, H.-R. Kuang, S. Radia, and R. Chansler, “The hadoop distributed file system,” in *Proceedings of the 26th IEEE Symposium on Mass Storage Systems and Technologies*, Apr 2007.

[12] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer Verlag, 1995.

[13] Y. Zhao, X. Liu, S. Gan, and W. Zheng, “Predicting disk failures with HMM- and HSMM-based approaches,” in *Proceedings of the 10th industrial conference on Advances in data mining: applications and theoretical aspects*, July 2010, pp. 390–404.