Predicting Hard Drive Failures for Cloud Storage Systems

Dongshi Liu, Bo Wang, Peng Li, Rebecca J. Stones, Trent G. Marbach, Gang Wang, Xiaoguang Liu, Zhongwei Li*

Nankai-Baidu Joint Laboratory, College of Computer Science, Nankai University, Tianjin, China
{liudongshi, wangb, lipeng, rebecca.stones82, trent.marbach, wgzwp, liuxg, lizhongwei}@nbjl.nankai.edu.cn

Abstract. To improve reactive hard-drive fault-tolerance techniques, many statistical and machine learning methods have been proposed for failure prediction based on SMART attributes. However, disparate datasets and metrics have been used to experimentally evaluate these models, so a direct comparison between them cannot readily be made.
In this paper, we provide an improvement to the Recurrent Neural Network model, which experimentally achieves a 98.06% migration rate and a 0.0% mismigration rate, outperforming the state-of-the-art Gradient-Boosted Regression Tree model, and achieves 100.0% failure detection rate at a 0.02% false alarm rate, outperforming the unmodified Recurrent Neural Network model in terms of prediction accuracy. We also experimentally compare five families of prediction models (nine models in total), and simulate the practical use.

1 Introduction

Modern cloud storage systems and other large-scale data centers often host hundreds of thousands of hard drives as their primary data storage device. While the theoretical annual failure rate of a single hard drive is low, in such large numbers they are a primary source of failure in today's cloud storage systems [22,23]. Hard-drive failure leads to service unavailability, which negatively impacts the user experience, and can cause permanent data loss.

Modern hard drives incorporate Self-Monitoring, Analysis and Reporting Technology (SMART) [1], but SMART attributes cannot directly provide satisfactory hard-drive failure prediction performance [15]. As a result, many statistical and machine learning methods utilize SMART attributes to substantially improve upon hard-drive failure prediction performance [2, 4–16, 18–21, 24–31]. However, differences in experimental setups make it hard to compare their respective performances and determine which model is most effective.

Difficulties comparing model performance hinder the realistic application in cloud storage systems. Two major differences are the choice of experimental dataset and the choice of experimental metric. A wide variety of datasets have been used; we tabulate them in Table 1 below[1]. Basic statistics of the datasets in this paper are listed in Table 3.

Table 1. Datasets Used in Previous Work

Dataset		Reference(s)
no. drives	no. failed drives	
1,936	9	[8]
3,744	36	[9]
369	191	[14, 15, 20, 24–26, 30]
23,395	433	[10–12, 16, 28, 31]
38,989	170	[11, 12, 28]
10,157	147	
Backblaze data center datasets		[3, 4, 13]

Most prior work [2, 4, 5, 7–10, 12–15, 18–21, 24–26, 30, 31] uniformly treated hard-drive failure prediction as a binary classification problem, and evaluated the model performance in terms of *failure detection rate* (FDR), defined as the proportion of failed drives that are correctly classified as failed, and *false alarm rate* (FAR), defined as the proportion of good drives that are incorrectly classified as failed. Some previous work [10, 12, 20, 24, 25] also incorporate *time in advance* (TIA), which is defined as the mean time between predicted failure and actual failure.

Instead of binary classification, some prediction models [16, 28] predict the residual life of hard drives, described by a drive's *health degree*. In this context, a drive's residual life is ordinarily predicted to fall into an interval, and prediction accuracy is measured by the number of predictions falling into the correct interval.

Li et al. [11] recently proposed two new performance metrics for hard drive failure prediction models: the *migration rate* (MR), defined as the proportion of data that is successfully migrated before its disk failed, and the *mismigration rate* (MMR), defined as the proportion of data on healthy disks that is migrated needlessly. Along with the traditional metrics (FDR, FAR, and TIA), we make use of these stricter metrics in this paper.

Four other significant issues that hinder model comparison are the following: (a) small datasets may be insufficient for adequately training the models, potentially leading to under-evaluated prediction performance results [8, 9, 14, 15, 24–26, 30] and contain too few failed drives, which negatively impacts both training and experimentation:

[1] Here and throughout the paper, despite the grammatical mismatch between "good" vs. "failed", for brevity we use "failed" as an adjective to describe hard drives which fail during data collection; all other hard drives are "good". This awkward nomenclature is consistent with many papers on this topic.

> ... detailed studies of very large populations are the only way to collect enough failure statistics to enable meaningful conclusions — Pinheiro et al., 2007 [17];

(b) different authors have chosen varying sets of SMART attributes to include and exclude in their experimental evaluations; (c) partitioning drives into test, training, and (possibly) validation sets has been done in various ways; and (d) some datasets [14, 15, 20, 24–26, 30] can be regarded as obsolete: their SMART information is inconsistent with the current SMART standard.

The main contributions of this paper are as follows:

- *Recurrent Neural Network model improvements.* We optimize the Recurrent Neural Network (RNN) model in terms of MR and MMR. We define a four-layer network model in which the output layer contains two additional nodes (MR and MMR). Further, during training, samples are instead considered over the entire life of a typical hard drive.
- *Nine models from five families.* We experimentally compare the performance of nine hard-drive failure prediction models on the same datasets collected from a real-world data center, using metrics MR and MMR, along with the traditional metrics FDR, FAR, and TIA.
- *Data center simulations.* We continue experimentation on six reasonable models by simulating their use on various drive families, in small-scale data centers and data centers with a mixture of drive models.

The paper is organized as follows: In Section 2, we survey related work on hard-drive failure prediction using SMART attributes. Section 3 presents the modified RNN model. Section 4 first gives a description of the datasets and how they are preprocessed, then presents the experimental results. We summarize the impact of this work in Section 5.

2 Related Work

SMART is a monitoring system which is widely used in modern hard drives. However, a simple SMART threshold-based method for hard-drive failure prediction results in an impractically poor FDR of around 3–10% when achieving a suitably low FAR around 0.1% [15]. Here we survey the methods used to overcome this problem.

The majority of prior work considered hard-drive failure prediction as a binary classification problem, including Bayesian approaches [4, 8], the Wilcoxon rank-sum test [9,14,15], a support vector machine method [15,31], hidden Markov models [30], a method involving Mahalanobis distance [24–26], backpropagation artificial neural networks [31], a classification tree model [10], a regularized greedy forest model [2], a Gaussian mixture model [20], an online random forest model [27], and a method using the FastTree algorithm [29].

Realistically, hard drives deteriorate gradually, so some previous work instead studied "health degree" prediction. Li et al. [10] proposed a Regression Tree model and defined a hard drive's health degree as its failure probability. Pang et al. [16] and Xu et al. [28] treated health degrees as the remaining working time of a hard drive before an actual failure occurs. The accuracy of health degree prediction was used to test a combined Bayesian network model [16] and a recurrent neural network model [28]. Li et al. [11, 12] improved the gradient-boosted regression tree (GBRT) method for hard-drive failure prediction.

From among these various methods, we select the most competitive to compare with our modified RNN model experimentally, which we describe in detail in the next section. Except for [11], all prior experimental evaluations did not incorporate data migration, which is a more critical factor for large-scale storage systems, such as cloud storage systems. One of the major motivations of this paper is thus to reevaluate these methods in an identical and up-to-date setting.

3 Extended RNN Model

We modify the network structure as depicted in Figure 1. In [28], the previous time step's hidden layer is fed into the current hidden layer, whereas we feed in not only the previous time step's hidden layer but also the previous time step's output layer; this is a main distinction between the two methods.

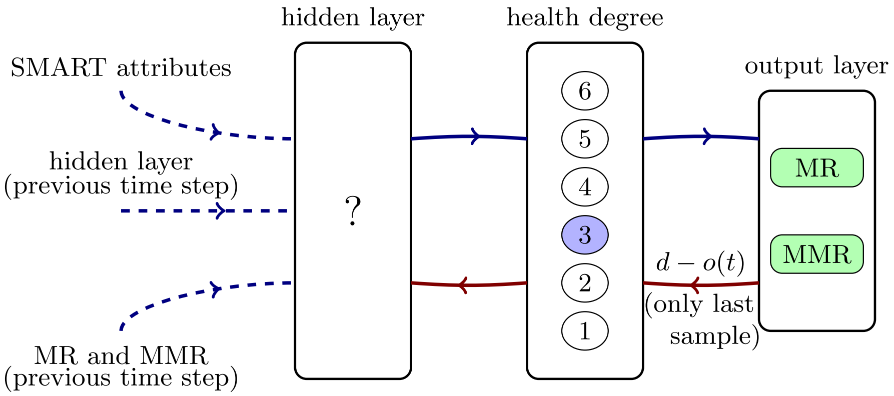

Fig. 1. The modified RNN model at time t; the SMART attributes at time t, hidden layer and output from time $t-1$ is fed into the hidden layer. It has four layers: an input layer, a hidden layer, a health-degree layer, and an output layer. We proceed drive by drive, then SMART-attribute sample by sample. MR is calculated from failed drives, and MMR is calculated from good drives.

When training RNN models, we incorporate the entire life of hard drives in the training set. The health-degree layer uses the softmax function to ensure the values of the six nodes form a valid probability distribution (i.e., all values are greater than 0 and their sum is 1). Each node's label represents the residual-life level a sample maps into. In the training process, for each sample we choose the node with the maximum value (i.e., the residual-life level a sample most likely maps into) in the health-degree layer. Then we calculate MR or MMR at this time step assuming the migration rate in the corresponding level as in [11], and update the MR or MMR value until the last sample of a hard drive.

Algorithm 1 Modified RNN model training procedure

```
Input: samples for all hard drives, a four-layer RNN model with initial weight matrices
Output: network weight matrices
for hard drive $D$ do
    for each SMART sample for hard drive $D$ do
        compute the hidden layer and health degree probabilities (health-degree layer)
        as per an unmodified three-layer RNN as per [28], subjoining the previous time
        step's output layer as input
        find the node in the health-degree layer with the maximum value (i.e., probability), and calculate MR or MMR of the current SMART sample
        if not the last sample of a hard drive then
            update the MR and MMR output $o(t)$ at the current time step $t$
            feed back as unmodified three-layer RNN as per [28] (i.e., excluding $d-o(t)$),
            subjoining the previous time step's output layer
        else
            feed back using the four-layer RNN and reset MR and MMR (i.e., including
            $d - o(t)$)
        end if
    end for
end for
return network weight matrices
```

When feeding back into the network, for each sample, we feed back the network excluding the output layer as in [28], subjoining the previous time step's output layer. If it is the last sample of a hard drive, we also feed back the gradient of the output layer and then reset MR and MMR to 0. The gradient of the output layer is $d - o(t)$ where $o(t)$ is the assessed MR or MMR value at time t, and d is the target values of MR and MMR, namely 1 and 0, respectively. Algorithm 1 gives the details for training the modified RNN prediction model.

When testing, for each SMART record, we feed forward the network using weight matrices obtained from the training process. If the node with the maximum value in the health-degree layer falls into a level 1 through 5, the record is labeled as failed, otherwise good. Then we calculate FDR, FAR, and TIA like other binary classifiers. Metrics MR and MMR are calculated as in the training process.

4 Experimental Results

We test the modified RNN model and eight other hard-drive failure prediction methods, which divide into five categories, tabulated in Table 2. We evaluate the various hard-drive failure prediction methods in terms of FDR, FAR, TIA, MR, and MMR, on the datasets described in Section 4.1. When measuring FDR, FAR, and TIA, we apply a voting-based failure detection algorithm [31]. When measuring MR and MMR, we process the samples sequentially for each hard drive, like in [11].

Table 2. Models we evaluate in this paper

Model family	Name	Year	Reference(s)
Probabilistic	Naive Bayes classifier	2001	[8]
	Bayesian network	2016	[4]
Support Vector Machine (SVM)		2005+	[15, 31]
Decision Tree	CT RT (part of CART)	2014	[10]
Boosting	GBRT	2016+	[11, 12]
	XGBoost	new	
Time series	HMM	2010	[30]
	RNN	2016	[28]

4.1 Dataset Description and Preprocessing

Datasets and Preprocessing Our datasets are from two real-world data centers. The data from the first data center, called dataset W, was released in [31]. The data from the second data center, called datasets M and S, were first used in [28]. The details of the three datasets are listed in Table 3.

Table 3. Dataset Statistics

Dataset	Class	No. disks	No. samples
W	good	22,962	3,837,568
	failed	433	158,150
S	good	38,819	5,822,850
	failed	170	97,236
M	good	10,010	1,681,680
	failed	147	79,698

We use three non-parametric statistical methods—reverse arrangement test, rank-sum test, and z-scores [15] to select features as the SMART attributes in our datasets are non-parametrically distributed (which is consistent with the observations in [9, 15]). We list the selected features for dataset W in Table 4 and for datasets M and S in Table 5. The differences between SMART attributes selected in Table 4 and Table 5 are because drives of data sets W and M, S are from different data centers and drive models, with different collected SMART attributes. We divide the three datasets, taking 70% of data as training set, 15% as validation set, and 15% as test set.

When using the RNN model, we map all input data to $[0, 1]$, which we do by replacing the original value x of a feature by

$$x \mapsto \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

where $x_{\max}$ and $x_{\min}$ are the maximum and minimum values of this feature in the training set, respectively. To leverage the relatively long sequence of historical information of SMART attributes, we sample one SMART record in each 24-hour period, consistent with [28].

Table 4. SMART features for dataset W

ID	Attribute name	Type
1	Raw Read Error Rate	basic, change rate
2	Spin Up Time	basic
3	Reallocate Sectors Count	basic
4	Seek Error Rate	basic
5	Power On Hours	basic
6	Reported Uncorrectable Errors	basic
7	High Fly Writes	basic
8	Temperature Celsius	basic
9	Hardware ECC Recovered	basic, change rate
11	Reallocated Sectors Count (raw value)	basic, change rate

Table 5. SMART features for datasets M and S

ID	Attribute name	Type
1	Raw Read Error Rate	basic, change rate
2	Spin Up Time	basic
3	Reallocate Sectors Count	basic, change rate
4	Seek Error Rate	basic
5	Power On Hours	basic
8	Temperature Celsius	basic
10	Current Pending Sector Count (raw value)	basic

4.2 Dataset W

For dataset W, Figure 2 plots the FDR and FAR using voting-based failure detection of each of the 9 methods. We see that increasing the number of voters generally reduces the FAR. Based on this experiment, in the subsequent experiments, we set the number of voters $N = 7$; there does not appear to be any significant benefit to choosing a greater N value. Not included in the figure are the TIA measurements; for $N = 7$, they ranged from 217 hours (HMM) to 264 hours (SVM), none of which would be problematic in practice. Excluding the naive Bayes classifier and the HMM model, all of the models give 24+ hours warning for 90%+ of the drives and 72+ hours warning for 84%+ of the drives.

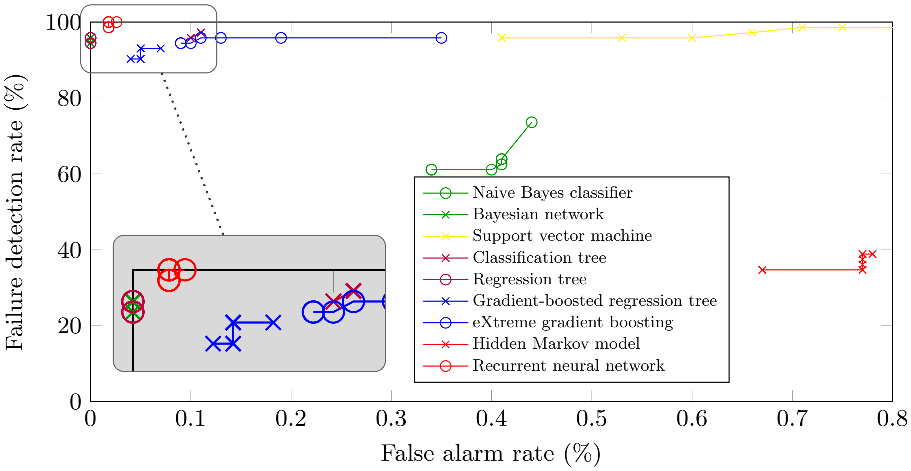

Fig. 2. Impact of the voting-based method on prediction accuracy (FDR vs. FAR) of the various prediction models, as the number of voters N varies in $\{1, 3, 5, 7, 11, 17, 27\}$ as in [10]; dataset W. The FAR generally decreases as N increases, so the plots are from right (small N) to left (large N). Some plots appear to have fewer than seven points because the FDR and FAR does not always vary with N in these experiments.

In Figure 2, we see that the results of the Bayesian network model outperforms the naive Bayes classifier, indicating that SMART attributes not following Gaussian distributions, which agrees with the observations in [9, 15].

Comparing the decision-tree and boosting models (CT, RT, GBRT, and XGBoost) using Figure 2, we observe that all four consistently achieve high FDR (93%+ when $N = 7$). Moreover, CT, RT, and GBRT achieve a small false alarm rate (when $N = 7$, the maximum observed is in the CT model at 0.11%).

In Figure 2, the SVM model achieves 97.22% FDR with 0.66% FAR (when $N = 7$), which is significantly better than the results observed in [15, 31]. We attribute this discrepancy primarily to two factors: a different choice of SMART attributes than in [31], and a much larger dataset than in [15] (369 hard drives; 191 failed). These results highlight the importance of re-testing the various methods on an "equal playing field".

The results for the HMM are the worst among those in Figure 2; this level of prediction accuracy would render it impractical for use in a real-world setting. Curiously, if we only use the SMART attribute "Raw Read Error Rate" to build the HMM, the model achieves 90.27% FDR at 0% FAR, outstripping its results in Figure 2. This result indicates that the HMM is better suited to a small set of attributes and performs poorly when using multi-dimensional attributes. When we use the SMART attribute “Reallocate Sectors Count”, HMM only achieves 36.11% FDR at 0.02% FAR, which illustrates HMM does not always achieve high FDR when using a single SMART attribute, so its prediction accuracy varies according to different choice of SMART attributes.

The modified RNN model consistently achieves both a high FDR of 100.0% and a low FAR of 0.02% (when $N = 7$), and Figure 2 indicates it outperforms all models except the Bayesian network and RT models, which achieve a fractionally better FAR of 0.00%, but a fractionally worse FDR of 96% (when $N = 7$).

We further compare the modified RNN model with the unmodified model in Table 6. The modified RNN model outperforms the unmodified RNN model in terms of both prediction accuracy and data migration. The MR increases by more than 18 percentage points. When data fails to migrate, there is a heightened risk of data loss: it takes fewer additional failures to lose data permanently. Furthermore, in erasure-coded cloud storage systems, if migration is incomplete before a hard drive fails, we trigger reconstruction, consuming system resources.

Table 6. The unmodified vs. modified RNN model, along with the RT, GBRT and XGBoost models

Model	FDR (%)	FAR (%)	TIA (h)	MR (%)	MMR (%)
unmodified RNN	95.83	0.03	255	79.92	0.01
modified RNN	**100.0**	0.02	**263**	**98.06**	**0.00**
RT	95.83	**0.00**	262	91.31	0.15
GBRT	93.06	0.05	259	94.62	0.07
XGBoost	95.83	0.11	262	94.65	0.03

Experimental results for the RT, GBRT, XGBoost, and RNN models are included in Table 6. The GBRT and XGBoost models outperforms the RT model using the data migration metrics (MR and MMR), but the opposite is true for prediction accuracy metrics (FDR, FAR, and TIA). We also observe that the modified RNN model outperforms the RT, GBRT, and XGBoost models in terms of MR and MMR.

The motivation behind introducing MR and MMR in [11] was that FDR, FAR, and TIA are sometimes misleading in practice, and these observations further support this claim. High prediction accuracy does not necessarily imply more appropriate data migration is taking place, and thus does not necessarily imply a more reliable system.

4.3 Simulating Practical Use

We evaluate model performance by simulating their practical use in real-world data centers: (a) being used with different hard drive families, (b) being used in small-scale datasets, and (c) being used with a mixture of drive models. We exclude the naive Bayes classifier and the HMM, due to their poor performance. We also abandon the SVM model due to its higher FAR on dataset W compared with the remaining six models.

Table 7. Prediction and migration accuracy on datasets M and S

Model	M		S	
	FDR(%)	FAR(%)	FDR(%)	FAR(%)
Bayesian network	95.56	0.69	92.16	0.36
CT	95.45	0.57	96.15	0.52
RT	93.33	0.74	94.11	0.13
GBRT	86.36	0.15	84.62	0.06
XGBoost	95.45	0.55	96.15	0.14
RNN	**100.0**	**0.02**	**100.0**	**0.01**

Model	M		S	
	MR(%)	MMR(%)	MR(%)	MMR(%)
RT	95.45	0.23	91.00	0.04
GBRT	91.76	0.09	93.08	**0.02**
XGBoost	95.45	0.12	93.33	0.07
RNN	**98.75**	**0.03**	**98.58**	**0.02**

Datasets M and S Hard drive models, manufacturers, and other environmental factors influence the statistical behavior of failures [22]. Even if made by the same manufacturers, different hard drive models have different characteristics, which may influence their reliability. Therefore, effectiveness on various hard drive models is an important factor in prediction models. With this motivation, we evaluate the six remaining models on datasets M and S, whose hard drive models are different to dataset W. Experimental results are tabulated in Table 7.

Comparing Table 7 to Figure 2, we observe changes in the FAR for the Bayesian network model (from 0.00% to 0.36%+), the CT model (from 0.11% to 0.52%+), and the RT model (from 0.00% to 0.13%+). We also observe that the GBRT model has an FDR of around 93% for the W dataset (FAR 0.05%), whereas it is around 86% for the M dataset (FAR 0.15%) and around 85% for the S dataset (FAR 0.06%). On the M and S datasets, the modified RNN model outperforms other models according to the metrics FDR, FAR, MR, and MMR.

Small-scale Datasets The datasets W, S, and M all contain a large number of hard drives. However, in real-world data centers, prediction models may be used on small or medium-sized datasets. We compare the remaining models on three “synthetic” datasets, like in [10,11], named A, B, C, by randomly choosing 10%, 25%, and 50% of all the good and failed hard drives respectively from dataset W. Table 8 tabulates the experimental results on these small-scale datasets.

Table 8. Prediction and migration accuracy on synthetic small-scale datasets

Model	A		B		C	
	FDR(%)	FAR(%)	FDR(%)	FAR(%)	FDR(%)	FAR(%)
Bayesian network	98.61	0.76	98.61	0.08	98.61	0.82
CT	98.61	0.38	97.22	0.20	98.61	0.13
RT	97.22	0.31	95.83	0.04	95.83	**0.00**
GBRT	76.39	**0.01**	84.72	**0.02**	90.28	0.04
XGBoost	100.0	0.08	97.22	0.08	95.83	0.12
RNN	**100.0**	0.02	**100.0**	0.04	**100.0**	0.03

Model	A		B		C	
	MR(%)	MMR(%)	MR(%)	MMR(%)	MR(%)	MMR(%)
RT	84.82	0.79	91.74	0.04	92.70	0.09
GBRT	93.18	0.66	92.99	0.07	94.31	0.06
XGBoost	89.37	0.12	90.40	**0.01**	91.14	**0.01**
RNN	**98.06**	**0.40**	**97.89**	0.24	**98.06**	0.30

In Table 8, we observe only minor performance degradation as the size of the dataset decreases, although the FDR for GBRT drops from around 90% to around 76%. The modified RNN model outperforms the others in terms of FDR and MR, while the metrics FAR and MMR do not strongly favor a method.

Non-ideal Datasets A real-world data center often has many engine rooms containing multiple hard-drive models. Though building a distinct prediction model for each hard drive model would achieve better results, this is impractical due to the time spent on data collection. To experimentally evaluate a non-ideal setup, we simulate two situations that might arise in a data center:

1. we have a large number of drives in the same drive family, together with different drive families whose data are insufficient for building models; and
2. we have multiple drive families with individually insufficient data, but together provide sufficient data.

For the first case, we build models using the dataset M and test model performance using dataset S (denoted M→S) or vice versa (denoted S→M). For the second case, we create a mixed dataset (denoted MS) by merging 25% of hard drives from the M and S datasets. We build models using the dataset MS and test model performance using datasets M, S, and MS. The results are denoted MS→S, MS→M, and MS→MS. We do not use dataset W here because the number of SMART attributes is inconsistent with the datasets M and S.

Table 9 tabulates the experimental results for M→S and S→M, and Table 10 tabulates the experimental results for MS→M, MS→S, and MS→MS.

Table 9. Prediction and migration accuracy for M→S and S→M

Model	S→M		M→S	
	FDR(%)	FAR(%)	FDR(%)	FAR(%)
Bayesian network	**100.0**	99.63	**100.0**	99.25
CT	77.27	92.79	69.23	67.99
RT	77.27	92.56	50.00	8.61
GBRT	77.27	76.55	61.54	23.72
XGBoost	77.27	92.78	73.08	92.03
RNN	**100.0**	**0.20**	**100.0**	**0.00**

Model	S→M		M→S	
	MR(%)	MMR(%)	MR(%)	MMR(%)
RT	77.29	19.13	56.01	5.94
GBRT	77.27	29.17	69.56	7.12
XGBoost	77.29	19.08	60.08	6.44
RNN	**98.75**	**0.16**	**99.78**	**0.00**

In Table 9, the proposed RNN model is the only model which consistently achieves practicable performance for all metrics, other models all have impractically high FAR and high MMR. This may because RNN utilizes the long-term dependencies among SMART attributes and builds models according to the historical fluctuations of data, whereas the other five models are all based on numerical values. Though the data in datasets M and S have numerical differences, they may have similar historical fluctuations.

Since HMM also utilizes historical fluctuations of data, we perform an additional test for this model: we observe that HMM achieves 34.62% FDR at 6.31% FAR for M→S and 68.18% FDR at 0.00% FAR for S→M. We likewise test the unmodified RNN model, which achieves 76.47% FDR at 0.06% FAR for M→S and 37.78% FDR at 53.27% FAR for S→M. These observations are somewhat consistent with the hypothesis that long-term dependencies are responsible for the observations in Table 9, but there may be additional reasons for these results.

Importantly, the results in Table 9 strongly indicate how a hard-drive failure model trained for one drive model may be unusable for predicting failures in another model, and how an idealized experimental environment may exaggerate the effectiveness of hard-drive failure prediction. In these results, the difference is extreme: going from nearly 0% FAR to nearly 100% FAR.

The results of all six models for the experiments MS→S and MS→M in Table 10 are similar to or slightly worse than the corresponding results in Table 7 (where we use training and test data from the same dataset). These results indicate simply creating a mixed training set is a practical method to overcome the problem arising in Table 9. All six models have practicable results for MS→MS, yet again we see the proposed RNN model outperforming the others.

Table 10. Prediction and migration accuracy for dataset MS

Model	MS→M		MS→S		MS→MS	
	FDR(%)	FAR(%)	FDR(%)	FAR(%)	FDR(%)	FAR(%)
Bayesian network	95.56	0.77	92.16	0.34	94.79	0.45
CT	86.36	0.65	92.31	0.42	89.58	0.49
RT	90.91	1.09	92.31	0.77	91.67	0.80
GBRT	86.36	**0.16**	84.62	0.21	83.33	0.16
XGBoost	86.36	0.43	92.31	0.20	89.58	0.22
RNN	**100.0**	0.44	**100.0**	**0.00**	**100.0**	**0.01**

Model	MS→M		MS→S		MS→MS	
	MR(%)	MMR(%)	MR(%)	MMR(%)	MR(%)	MMR(%)
RT	90.93	0.26	88.20	0.04	89.48	0.14
GBRT	88.91	**0.16**	88.33	0.09	88.60	0.11
XGBoost	90.92	0.24	92.92	0.05	91.39	0.13
RNN	**100.0**	0.44	**100.0**	**0.00**	**99.22**	**0.00**

5 Conclusion

In this paper, we implement a modified RNN model, and evaluate eight other models from five families on real-world datasets using various metrics experimentally. As a result, we give a fairer comparison between the models, which shows that the modified RNN model consistently achieves the best or nearly the best experimental results among the methods studied.

While experiments mostly give rise to comparable results to their original authors, for the support vector machine method, we observe a far higher prediction accuracy than presented by their authors in [15, 31], which we attribute to their small-size dataset and a different choice in SMART attributes.

The traditional metrics (FDR, FAR, and TIA) can sometimes suggest that one method outperforms another, while after incorporating migration (MR and MMR), the opposite is true. An example of this is the surprising observation that the RT method outperforms the GBRT and XGBoost methods on the traditional metrics, but in Table 6, we see that GBRT and XGBoost methods outperform the RT method in terms of MR and MMR.

In Section 4.2, we make the curious observation about how, for the hidden Markov model, prediction accuracy changes wildly depending on the selection of SMART attributes. In most work on this topic (including this paper), the authors make a selection of SMART attributes they consider most suitable. It would be interesting to expand this work to include the impact of the choice of SMART attributes, which we observe is significant in the example above.

We put forward the following advice when evaluating hard-drive failure prediction in cloud storage system:

- *Evaluation metric selection.* We observe that a high FDR does not necessarily imply a high MR. In a cloud storage system and other large-scale storage systems, we need to continuously migrate at-risk data, thereby consuming system resources. Thus migration-based metrics, such as MR and MMR, are better suited for evaluating model performance for cloud storage systems, than the less sophisticated metrics FDR and FAR.
- *Mixed drive models.* In Table 9, we make an observation that a model trained using data from one drive model may be useless at predicting hard-drive failure for a different drive model. An storage system operator should bear this in mind when training models for experimental evaluation. Further, in Table 10 we observe that this problem can be alleviated by using a training set that includes drives from both models. This is particularly relevant for cloud storage systems, which are likely to have multiple drive models.

Acknowledgments

This work is partially supported by NSFC (61872201, 61702521, 61602266, and U1833114); STDP of Tianjin (17JCYBJ15300, 16JCYBJ41900, 18ZXZNG00140, and 18ZXZNGX00200); and Fundamental Research Funds for Central Universities, KLMDASR.

References

1. Allen, B.: Monitoring hard disks with SMART. Linux J. (117), 74–77 (2004)
2. Botezatu, M.M., Giurgiu, I., Bogojeska, J., Wiesmann, D.: Predicting disk replacement towards reliable data centers. In: Proc. SIGKDD. pp. 39–48 (2016)
3. Chaves, I.C., de Paula, M.R.P., Leite, L.G.M., Gomes, J.P.P., Machado, J.C.: Hard disk drive failure prediction method based on a bayesian network. In: Proc. IJCNN (2018)
4. Chaves, I.C., de Paula, M.R.P., Leite, L.G., Queiroz, L.P., Gomes, J.P.P., Machado, J.C.: BaNHFaP: A Bayesian network based failure prediction approach for hard disk drives. In: Proc. BRACIS. pp. 427–432 (2016)
5. Ganguly, S., Consul, A., Khan, A., Bussone, B., Richards, J., Miguel, A.: A practical approach to hard disk failure prediction in cloud platforms: Big data model for failure management in datacenters. In: Proc. BigDataService. pp. 105–116 (2016)
6. Garcia, M., Ivanov, V., Kozar, A., Litvinov, S., Reznik, A., Romanov, V., Succi, G.: Review of techniques for predicting hard drive failure with smart attributes. International Journal of Machine Intelligence and Sensory Signal Processing **2**(2), 159–172 (2018)
7. Goldszmidt, M.: Finding soon-to-fail disks in a haystack. In: Proc. HotStorage (2012)
8. Hamerly, G., Elkan, C.: Bayesian approaches to failure prediction for disk drives. In: Proc. ICML. pp. 202–209 (2001)
9. Hughes, G.F., Murray, J.F., Kreutz-Delgado, K., Elkan, C.: Improved disk-drive failure warnings. IEEE Trans. Rel. **51**(3), 350–357 (2002)
10. Li, J., Ji, X., Jia, Y., Zhu, B., Wang, G., Li, Z., Liu, X.: Hard drive failure prediction using classification and regression trees. In: Proc. DSN. pp. 383–394 (2014)

11. Li, J., Stones, R.J., Wang, G., Li, Z., Liu, X., Xiao, K.: Being accurate is not enough: New metrics for disk failure prediction. In: Proc. SRDS. pp. 71–80 (2016)
12. Li, J., Stones, R.J., Wang, G., Liu, X., Li, Z., Xu, M.: Hard drive failure prediction using decision trees. Reliab. Eng. Syst. Saf. **164**, 55–65 (2017)
13. Mahdisoltani, F., Stefanovici, I., Schroeder, B.: Proactive error prediction to improve storage system reliability. In: Proc. USENIX ATC. pp. 391–402 (2017)
14. Murray, J.F., Hughes, G.F., Kreutz-Delgado, K.: Hard drive failure prediction using non-parametric statistical methods. In: Proc. ICANN/ICONIP (2003)
15. Murray, J.F., Hughes, G.F., Kreutz-Delgado, K.: Machine learning methods for predicting failures in hard drives: A multiple-instance application. J. Mach. Learn. Res. **6**, 783–816 (2005)
16. Pang, S., Jia, Y., Stones, R., Wang, G., Liu, X.: A combined Bayesian network method for predicting drive failure times from SMART attributes. In: Proc. IJCNN. pp. 4850–4856 (2016)
17. Pinheiro, E., Weber, W.D., Barroso, L.A.: Failure trends in a large disk drive population. In: Proc. FAST (2007)
18. Pitakrat, T., van Hoorn, A., Grunske, L.: A comparison of machine learning algorithms for proactive hard disk drive failure detection. In: Proc. SIGSoft symposium on Architecting critical systems. pp. 1–10 (2013)
19. Qian, J., Skelton, S., Moore, J., Jiang, H.: P3: Priority based proactive prediction for soon-to-fail disks. In: Proc. NAS. pp. 81–86 (2015)
20. Queiroz, L.P., Rodrigues, F.C.M., Gomes, et al.: A fault detection method for hard disk drives based on mixture of Gaussians and nonparametric statistics. IEEE Trans Ind. Informat. **13**(2), 542–550 (2017)
21. Rincón, C.C.A., Pâris, J.F., Vilalta, R., Cheng, A.M., Long, D.D.: Disk failure prediction in heterogeneous environments. In: Proc. SPECTS. pp. 1–7 (2017)
22. Schroeder, B., Gibson, G.A.: Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you? In: Proc. FAST. vol. 7, pp. 1–16 (2007)
23. Vishwanath, K.V., Nagappan, N.: Characterizing cloud computing hardware reliability. In: Proc. SoCC. pp. 193–204 (2010)
24. Wang, Y., Ma, E.W., Chow, T.W., Tsui, K.L.: A two-step parametric method for failure prediction in hard disk drives. IEEE Trans Ind. Informat. **10**, 419–430 (2014)
25. Wang, Y., Miao, Q., Ma, E.W., Tsui, K.L., Pecht, M.G.: Online anomaly detection for hard disk drives based on Mahalanobis distance. IEEE Trans. Rel. **62**, 136–145 (2013)
26. Wang, Y., Miao, Q., Pecht, M.: Health monitoring of hard disk drive based on Mahalanobis distance. In: Proc. PHM-Shenzhen. pp. 1–8 (2011)
27. Xiao, J., Xiong, Z., Wu, S., Yi, Y., Jin, H., Hu, K.: Disk failure prediction in data centers via online learning. In: Proc. ICPP. p. 35 (2018)
28. Xu, C., Wang, G., Liu, X., Guo, D., Liu, T.Y.: Health status assessment and failure prediction for hard drives with recurrent neural networks. IEEE Trans. Comput. **65**(11), 3502–3508 (2016)
29. Xu, Y., Sui, K., Yao, R., Zhang, H., Lin, Q., Dang, Y., Li, P., Jiang, K., Zhang, W., Lou, J.G., et al.: Improving service availability of cloud systems by predicting disk error. In: Proc. USENIX ATC. pp. 481–494 (2018)
30. Zhao, Y., Liu, X., Gan, S., Zheng, W.: Predicting disk failures with HMM- and HSMM-based approaches. In: Proc. Industrial Conference on Data Mining. pp. 390–404 (2010)
31. Zhu, B., Wang, G., Liu, X., Hu, D., Lin, S., Ma, J.: Proactive drive failure prediction for large scale storage systems. In: Proc. MSST. pp. 1–5 (2013)