2016 IEEE 35th Symposium on Reliable Distributed Systems

Being Accurate is Not Enough: New Metrics for Disk Failure Prediction

Jing Li[1], Rebecca J. Stones[1], Gang Wang[1*], Zhongwei Li[1], Xiaoguang Liu[1*], Kang Xiao[2]
[1]Nankai-Baidu Joint Lab, College of Computer and Control Engineering, Nankai University, Tianjin, China
[2] Qihoo 360 Technology Company, Beijing, China
Email: {lijing, rebecca.stones82, wgzwp, lizhongwei, liuxg}@nbjl.nankai.edu.cn, xiaokang@360.cn

***Abstract*—Traditionally, disk failure prediction accuracy is used to evaluate disk failure prediction model. However, accuracy may not reflect their practical usage (protecting against failures, rather than only predicting failures) in cloud storage systems.**

In this paper, we propose two new metrics for disk failure prediction models: migration rate, which measures how much at-risk data is protected as a result of correct failure predictions, and mismigration rate, which measures how much data is migrated needlessly as a result of false failure predictions. To demonstrate their effectiveness, we compare disk failure prediction methods: (a) a classification tree (CT) model vs. a state-of-the-art recurrent neural network (RNN) model, and (b) a proposed residual life prediction model based on gradient boosted regression trees (GBRTs) vs. RNN. While prediction accuracy experiments favor the RNN model, migration rate experiments can favor the CT and GBRT models (depending on transfer rates). We conclude that prediction accuracy can be a misleading metric. Moreover, the proposed GBRT model offers a practical improvement in disk failure prediction in real-world data centers.

I. Introduction

Nowadays, large scale data centers can host hundreds of thousands of servers, which often deploy hard disks as primary data storage device. There are many challenges facing data center management [1]–[4]. While a failure in a single disk might be rare, a system with thousands of disks will often experience failures and even simultaneous failures [5]–[7]. Disk failure can not only lead to service unavailability and hurt the user experience, but also result in permanent data loss. Therefore, high reliability is one of the biggest concerns in such systems.

Predicting disk failures before they actually occur allows us to handle them in advance, which can greatly enhance the storage system reliability. Most modern disks have Self-Monitoring, Analysis and Reporting Technology (SMART) [8], which monitors and compares disk attributes with preset thresholds, and issues warnings when attributes exceed the thresholds. However, SMART attributes alone can not reach a desirable prediction performance [9]. In order to improve prediction accuracy, a number of statistical and machine learning methods have been proposed to build disk failure prediction models based on SMART attributes [9]–[21].

Previous work [9]–[19] almost uniformly treats disk failure prediction simply as a yes/no problem, and are evaluated by measuring the *prediction accuracy* in terms of the *failure detection rate* (FDR) and *false alarm rate* (FAR). Later prediction models [20], [21] aim to predict the residual life of a disk, thereby enabling operators to allocate system resources more effectively for pre-warning processes while maintaining the quality of user services. The evaluation metrics used in these works are based on the classification accuracy; the possible disk residual life is partitioned into intervals, and the accuracy is measured in terms of the number of predictions that fall into the correct level. This is called an *accuracy of residual life level assessment* (ACC), and can be made for all failed/good samples [20] and for all failed/good disks [21].

Above all, the metrics used in previous work focus on the prediction models themselves, and isolate them from their application—storage systems, and especially cloud storage systems, where migration might be a continuous resource drain. With all else being equal, a higher prediction accuracy will always be beneficial, but practically, improving prediction accuracy incurs a trade-off in other ways, such as mean warning time, i.e., *time in advance* (TIA). For example, a disk failure prediction model may be improved to 100% prediction accuracy at the cost of reducing TIA, such as by reducing to one hour. In this case, the at-risk data could not be completely protected, even if they have been detected. As such, prediction accuracy does not give a complete picture.

In this paper, we introduce two performance metrics for disk failure prediction model: *migration rate* (MR), defined as the fraction of data on disks which go on to fail which is migrated, and *mismigration rate* (MMR), defined as the fraction of data on healthy disks that is migrated needlessly. In distributed storage systems, even if partial data on a failed disk is protected successfully, it is still valuable in practice, so we use the fraction of data rather than the number of disks that are migrated completely.

We also propose a residual life prediction model based on Gradient Boosted Regression Trees (GBRTs), which can result in more at-risk data being protected, less resources being wasted, and is more applicable to cloud storage system. On a dataset collected from two real-world data centers, we show that the new metrics (describing migration accuracy) are more meaningful than the previous ones (describing prediction accuracy) and the GBRT model outperforms the state-of-the-art disk residual life prediction model, Recurrent Neural Networks (RNNs), in terms of migration accuracy. We also offer improvements on the GBRT algorithm for this application, which are proposed based on the characteristics of disk residual life prediction.

1060-9857/16 $31.00 © 2016 IEEE
DOI 10.1109/SRDS.2016.17

The rest of the paper is organized as follows: In Section II, we survey related work of disk failure prediction and the evaluation metrics. Section III introduces the new metrics and models for disk residual life prediction. Section IV gives a description of the datasets and their curation. Experimental results are given in Section V, followed by conclusions in Section VI.

II. Related Work

In order to improve system reliability, researchers have focused on SMART-based proactive fault tolerance. With sufficient prediction accuracy, the technique can significantly reduce the negative impacts on system reliability and availability in the presence of failures.

To avoid false alarms, manufacturers set the thresholds conservatively to minimize FAR at the expense of FDR. The threshold-based algorithm implemented in disks can only obtain an FDR of around $3-10\%$ with a low FAR on the order of 0.1% [9]. So, to be useful, prediction accuracy has been improved in various ways, including Bayesian approaches [9], [10], using the Wilcoxon rank-sum test [11], [12], hidden Markov models [13], using the Mahalanobis distance [15], backpropagation artificial neural networks [14], and a classification tree method [17]. These works consider disk failure prediction as a binary classification issue (whether or not a disk is going to fail). Their goal is to detect as many at-risk disks as possible, while avoiding false alarms, measured using FDR, FAR, and TIA.

Realistically, disks do not deteriorate suddenly, but gradually. This is consistent with the long TIA observed in e.g. [14], [17]. Therefore, in the present authors' previous work, we proposed a disk health degree prediction model based on Regression Trees [17] where we defined a disk's health degree as its failure probability. A disk's health degree can be utilized to indicate trends in disk failure, allowing technicians to respond to warnings raised by the failure prediction model according to their health degrees. However, the health degree was defined as a probability in [17], which is not intuitive for pre-warning handling. Moreover, [17] did not use an explicit metric to evaluate the health degree models.

Recently, Pang *et al.* [21] proposed a new definition of health degree, which is defined as the remaining working time, *residual life*, of a disk before actual failure occurs. They implemented a Combined Bayesian Network (CBN) model, which combined the learning results from four individual classifiers using backpropagation artificial neural networks, evolutionary neural networks, support vector machines, and classification tree methods. They used *classification precision*, defined as accuracy of the health degree prediction for all test disks, to evaluate models. When adopting a division method, the CBN model could obtain over 60% prediction accuracy on failed disks.

Later, Xu *et al.* [20] considered the health status of disks had long-range dependency, and introduced a method based on Recurrent Neural Networks (RNN) to assess the health statuses of disks based on gradually changing sequential SMART attributes. They adopted a discrete classification method to define the levels of health status, which could indicate the residual life of disks. The evaluation metric used by them was the sample's prediction accuracy. Their ACC assessment took each testing sample as an input instant, and ignored the correlations among the samples from a single disk. On a large real-world dataset, their method achieved about $40\% \sim 60\%$ ACC on failed samples.

Both [21] and [20] treated prediction models as multiple classifiers, and the metrics used in them isolated the prediction models from their industrial applications. The ultimate goal of disk failure prediction is to avoid data loss, which not only requires predicting which disks are at risk, but also completing the pre-warning handling processes. To build more practical disk failure prediction model, the evaluation metric should consider the completion status for disk migration.

There have been some studies focusing on putting disk failure prediction into practice. Wu *et al.* [22] designed a proactive protection mechanism, IDO, which identifies at-risk disks and proactively migrates at-risk data of hot zones to a surrogate RAID set; RAIDSHIELD [18] was designed to use the joint failure probability to replace at-risk disks before their actual failures; and Fatman [23] migrated data from at-risk disks to reduce reconstruction costs when RS-decoding cold data. However, they simply migrated at-risk data raised by prediction models, without carefully allocating resources for migrations to reduce their impact on the quality of service. Given this, Ji *et al.* [24] employed the SSM (self-scheduling migration), the migration algorithm of which managed the priority of pre-warning handling processes in a reasonable way, to minimize the their impact on the performance.

In this paper, we propose new evaluation metrics for disk failure prediction models, and then propose a matching disk residual life prediction model by employing Gradient Boosted Regression Trees (GBRT), motivated by [21] and [20].

III. The Proposed Method

A. Motivation

In a proactive fault tolerance storage system, a disk failure prediction model runs in the background and monitors the disks in real time, periodically outputting their states (such as once an hour). When an alarm is raised by the model, the data on an at-risk disk is ordinarily migrated to other healthy disks immediately.

In typical storage systems (such as small disk arrays), which generally have small size, disk failures (and their corresponding alarms) are relatively rare. In this setting, we can allocate enough disks and bandwidth resources for all migration tasks without significantly reducing system availability. However, with the advent of cloud storage systems, hosting perhaps hundreds of thousands of servers, failures and even simultaneous failures occur frequently. Moreover, in such cloud storage systems, network and disk I/O have a great influence on the quality of service. To maintain high quality, operators should limit the system resources used for migration, but this increases the possibility of failing to protect the at-risk data.

The situation is worse with simultaneous failure predictions, with system resources competing with multiple migrations.

To evaluate a prediction model in terms of its migration accuracy, we consider the migration rate and mismigration rate (as defined in the introduction). These measures have the following properties: (a) They are meaningful and understandable, respectively describing the amount of at-risk data which is protected, and the amount of data which is protected needlessly, which affect the quality of service (reliability and availability). (b) They enable us to compare the quality of residual life prediction models in terms of migration accuracy.

In general, the higher the MR the better, and the lower the MMR the better. Higher MR means more at-risk data are protected successfully, thereby lesser data needs to be regenerated using other survived data, when failures occur. Since reconstruction will affect the performance of service seriously, the higher MR will improve more reliability and availability of systems. Lower MMR means less resources are wasted by incorrect failure predictions. For a given prediction model, we may have different values of MR and MMR in systems using different pre-warning handling strategies. In general, higher pre-warning migration transfer rates may induce higher values of MR and MMR. If all the migrations could be guaranteed to complete, MR and MMR are respectively equal to FDR and FAR.

B. Pre-warning Handling

When a failure is predicted, in order to minimize the impact on the quality of service for users, we wish to use as few resources as possible to protect the at-risk data. If the residual life of the at-risk disk can be predicted, we can adjust the transfer rate for migration accordingly.

Theoretically, if a disk containing exactly m TB of data will fail in exactly h hours, then there will be sufficient time to back up the data at m/h TBs per hour. This minimizes the bandwidth used for migration, although it requires that migration is completed just before the failure occurs. However, in practice, we instead have a failure prediction time of $\hat{h}$ hours, so we can back up the data at $m/\hat{h}$ TBs per hour, and, at a later point in time, the failure prediction time might have changed to $\hat{h}'$ hours and there will be m' TB of unmigrated data, so the transfer rate for migration can be updated to $m'/\hat{h}'$ TBs per hour. To avoid issues where m' or $\hat{h}'$ are very small, when the prediction is inaccurate, and having a continuously changing transfer rate as m' varies, we use predetermined transfer rates, as listed in Table I, and until the failure prediction time changes to a different level, the migration rate remains unchanged.

In Table I, we experiment with 3 different partitions, motivated by [24]. In each case, level 6 indicates that the disk works properly and does not need to be handled (with no migration). Levels 1 to 5 imply that the disk is predicted to fail, and its contents needs to be migrated, which is performed at the transfer rate specified in Table I. We choose shorter interval lengths and higher migration rates for shorter predicted residual life, as migrating data from these disks is more urgent. The goal of SMART by disk manufacturers is to provide 24 hours warning-time before disk failure [25], so level 1 transfer rates are set to complete within 24 hours. We just set the transfer rates casually and did not compare them with any other settings. It may be get better results with other carefully selected settings.

In the systems using binary classifiers to predict disk failure, all the warnings raised by prediction models could only be equally handled with (being migrated at an uniform transfer rate). Since a single detection can not give a confident prediction of a disk fault due to detection error, it is not appropriate to rashly predict that a disk is going to fail once it is classified as failed by the prediction model, without continuing to monitor it. When a warning raised by the model, it may seem reasonable to migrate data from it just during a prediction time interval, such as an hour, until all the data has been migrated. The migration will be continued, unless the disk is fluctuated to be predicted as a good one at the next prediction. The migrations, which are interrupted due to a contrary prediction, will continue to be processed, if the disks are predicted as at-risk disks again.

C. Prediction Model

In this paper, we propose a disk residual life prediction model based on Gradient Boosted Regression Trees (GBRTs) [26], which is more applicable to cloud storage systems. Each test has a quantitative target value describing the residual life of a disk rather than a class label indicating it as healthy or at-risk. GBRT is a gradient descent boosting technique based on tree averaging, and is an accurate and effective machine learning technique that can be used for both regression and classification problems. To avoid overfitting, the GBRT algorithm trains many tree stumps as weak learners, rather than full, high variance trees. Even the small trees have high bias. So a tree depth d (set to a small value) is used to control the size of trees.

We use regression trees as weak learners. Fig. 1 illustrates a simplified regression tree for disk residual life prediction. The SMART attributes are used as input vectors together with the target values representing residual life of disks. We begin at the root node (node 1), weighted as the mean (516.3 hours) of all samples in it on the target variable. This node is split based on the value of a SMART attribute "Power On Hours". If the value is ≤ 95, those samples move to node 2 (weighted 359.1 hours) and the other samples move to node 9 (weighted 910.2 hours). Nodes 2 and 9 split into two child nodes based on the value of "Reallocated Sectors Count (raw value)" in different ways. This process continues until the max depth of tree d is reached (in this case $d = 4$). Each leaf node is weighted with the mean of the residual life of its samples. The residual life of a disk is predicted as the weight of the leaf node.

The important difference between a binary classifier, such as the Classification Tree model [17], and the GBRT algorithms is how to set initial target values of training samples. In a binary classifier algorithm, the target value of every good sample is set to an uniform value (such as 1) and that of every failed

TABLE I: The three examples of migration transfer rate settings. The storage capacity of a disk is denoted by m.

Level	Partition 1		Partition 2		Partition 3	
	Residual life (hours)	Rate (per hour)	Residual life (hours)	Rate (per hour)	Residual life (hours)	Rate (per hour)
1	$0-24$	$m/5$	$0-48$	$m/12$	$0-72$	$m/24$
2	$25-72$	$m/24$	$49-96$	$m/48$	$73-144$	$m/72$
3	$73-168$	$m/72$	$97-192$	$m/96$	$145-240$	$m/144$
4	$169-336$	$m/168$	$193-336$	$m/192$	$241-360$	$m/240$
5	$337-500$	$m/336$	$337-500$	$m/336$	$361-500$	$m/360$
6	> 500	0	> 500	0	> 500	0

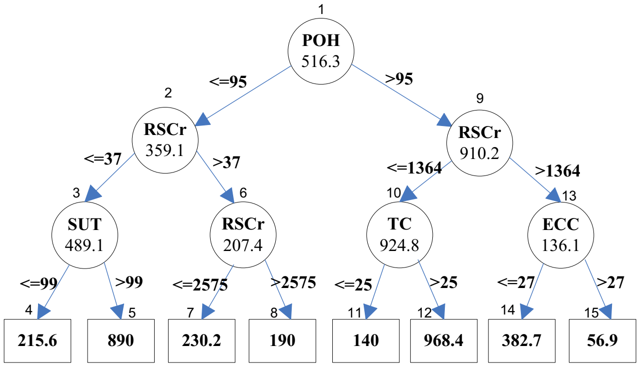

Fig. 1: An example of a regression tree for disk residual life prediction. Nodes are labeled 1 through 15, and the weights of leaf nodes give the predicted residual life in hours. SMART attributes: POH = Power On Hours, RSCr = Reallocated Sectors Count (raw value), TC = Temperature Celsius, SUT = Spin Up Time, ECC = Hardware ECC Recovered. The maximum possible node weight is 1000 hours, set for healthy disks.

sample is set to an another uniform value (such as -1). In the GBRT algorithm, the healthy samples are assigned a residual life of 1000 hours (but might actually remain healthy for much longer than 1000 hours), i.e. their target values are set to 1000 hours. For each failed sample, we set its target value to the disk's residual life (or, how long in advance it is before the disk fails). A higher value means better health condition. When the sample is collected at the moment the disk fails, its health degree is set to 0 hour.

To find the best split, the regression tree algorithm checks all possible splits (all values of the input SMART attributes) (a split could be e.g. whether POH ≤ 95). Determining the best split is achieved using the minimum of squares of nodes (instead of the usual greatest gain in information), namely

$$sq := \sum_j (y_j - \bar{y})^2, \tag{1}$$

where y_j is the remaining life of a disk based on the j-th sample, and $\bar{y} = \text{ave}_j(y_j)$. The sum (1) is over all disk samples that satisfy the splitting conditions of the ancestor nodes (e.g. at node 6 in the Fig. 1 example, the disks are precisely those that satisfy POH ≤ 95 and RSCr > 37).

For GBRTs, regression trees are introduced at each iteration (we use $c = 500$ iterations) to adjust for prediction errors (residuals) for each sample vs. the target value (the residual life) from the previous regression trees. The eventual aim of GBRTs is to fit the target value of each sample to its initial one (the true residual life). The residuals for each new tree are used to minimize the value of loss function, and then improve the quality of fit of each base learner.

The residuals of the i-th tree (used to determine the $(i+1)$-th tree) are given by

$$r^{(i+1)}[j] := r^{(i)}[j] - \alpha T^{(i)}[j] \tag{2}$$

where $T^{(i)}[j]$ is the prediction for the j-th sample from the i-th regression tree, α is a user-defined learning rate, and $r^{(1)}[j] = y_j$, and (1) is generalizes to

$$sq := \sum_j \left(r^{(i)}[j] - \overline{r^{(i)}}\right)^2. \tag{3}$$

We build GBRT models using SMART attributes and their change rates as input vectors together with the target values representing residual life of disks. Algorithm 1 gives the details for training the GBRT prediction model. When testing, a disk's residual life is predicted as the combined predictions by all the regression trees, or $\sum_{i=1}^{c} \alpha T^{(i)}[j]$.

As described in Section III-B, technicians only need to know which residual life interval a disk may fall into, and then they can appropriately handle with it (migrate data on it at an specified rate).

Therefore, at each iteration, the samples which have been predicted correctly (up to the intervals in Table I) by the previous regression trees need not to be used to train the following trees (i.e., the target residuals of which can be adjusted to 0). In this way, the new trees focus on the samples which have not been predicted into the correct interval by the existing predictors (a significant advantage of GBRT method). Specifically, for training, we change (2) such that

$$r^{(i+1)}[j] := 0 \tag{4}$$

whenever the current residual life prediction for a disk $\sum_{x=1}^{i} \alpha T^{(x)}[j]$ falls within the correct interval (the same interval that y_j belongs to).

If a disk's predicted residual life falls within a higher-level interval, at-risk data will be migrated at a higher rate, so the system reliability will be improved at cost of availability. We do not anticipate this being a significant cost, so if y_j falls

Algorithm 1 Training the GBRT model

```
Input: Training data set (including actual SMART attributes and residual life $y_j$), learning rate $\alpha$, number of regression trees $c$, tree depth $d$
Output: GBRTs $T^{(i)}$ used for predicting disk residual life
initialize $r^{(1)}[j] \leftarrow y_j$ for $j = 1$ to $n$
for regression tree $i = 1$ to $c$ do    $\triangleright$ build regression tree $T^{(i)}$ of depth $d$
    weight root node of $T^{(i)}$ with $\overline{r^{(i)}}$
    for $k = 1$ to $d$ do
        for each node $V$ at depth $k$ do
            for each possible split at $V$ do
                calculate $sq_{\mathrm{L}} + sq_{\mathrm{R}}$ from (3), where $L$ and $R$ are its two proposed child nodes
            end for
            split $V$ to minimize $sq_{\mathrm{L}} + sq_{\mathrm{R}}$
            weight $V$'s child nodes with $\mathrm{ave}_s(r^{(i)}[s])$, where the average is over all disks $s$ which satisfy the splitting conditions of its ancestor nodes
        end for
    end for
    update $r^{(i+1)}[j] \leftarrow r^{(i)}[j] - \alpha T^{(i)}[j]$ for $j = 1$ to $n$
end for
```

within levels 2, 3, 4, or 5 (in Table I), and the current residual life prediction for a disk $\sum_{x=1}^{i} \alpha T^{(x)}[j]$ is respectively within levels 1, 2, 3, or 4, we also change (2) such that

$$r^{(i+1)}[j] := 0. \tag{5}$$

IV. Dataset Description And Preprocessing

A. Datasets

Hard disk failures are complex in reality and do not follow a simple fail-stop model [27]. There are some types of disk failure, such as permanent whole-disk failures, transient performance problems and latent sector errors. However, in our paper, we focus on the permanent whole-disk failures, and regard the disks which are not connected permanently by the system as failed.

To test our new metrics and our model, we use a real-world dataset collected in two real-world data centers. Hourly samples were taken from working disks using smartmontools. Each sample contains all the SMART attribute values for a single disk at an exact time.

The data collected from the first data center, represented by "W", used in our previous work [14], contains 23,395 disks from an enterprise-class model, labeled "good" or "failed"[1]. For good disks, the samples in a week-long time period are recorded. Some samples may be missed because of sampling or storing errors. For each failed disk, samples in a period of 20 days before its actual failure were recorded. Some failed disks lost some samples if they did not survived 20 days of operation since we began to collect data.

We collect two additional datasets from a second data center, referred to as "S" and "M", used in [20] . The disks in the datasets are from two Seagate disk models (not the same as "W"). Hourly samples were recorded for each disk: for good disks, samples were taken over a week, and for failed disks, samples were taken over a 25-day period before failure occurred. Since some failed disks did not survive 25 days of operation since we began to collect data, so they had fewer samples. Table II lists the details of the three subsets of the data.

TABLE II: Details of the datasets.

Dataset	Class	No. disks	Period	No. samples
"W"	Good	22,962	7 days	3,837,568
	Failed	433	20 days	158,150
"S"	Good	38,819	7 days	5,822,850
	Failed	170	25 days	97,236
"M"	Good	10,010	7 days	1,681,680
	Failed	147	25 days	79,698

For every disk in the "W" dataset, there are 23 meaningful attributes in its SMART record. However, the values of some attributes are the same for good and failed disks and do not change during operation. So we filter them out and use only ten attributes to build our prediction models. In some cases, the raw values of some attributes are more sensitive to the health condition of disks. We select two raw values in addition to the ten normalized values to build our prediction models, giving the 12 *basic features* listed in Table III. For the disks in "S" and "M" datasets, since some features are not recorded, we only use the 7 attributes indicated in Table III.

TABLE III: Basic features (SMART attributes) used for the "W", "S", and "M" datasets.

ID #	Attribute Name	Datasets
1	Raw Read Error Rate	"W", "S", "M"
2	Spin Up Time	"W", "S", "M"
3	Reallocated Sectors Count	"W", "S", "M"
4	Seek Error Rate	"W", "S", "M"
5	Power On Hours	"W", "S", "M"
6	Reported Uncorrectable Errors	"W"
7	High Fly Writes	"W"
8	Temperature Celsius	"W", "S", "M"
9	Hardware ECC Recovered	"W"
10	Current Pending Sector Count	"W", "S", "M"
11	Reallocated Sectors Count (raw value)	"W"
12	Current Pending Sector Count (raw value)	"W"

B. Data Preprocessing

In our previous work [17], we calculated the absolute differences between the current values of the basic features and their corresponding values six hours prior as new features (called change features), and then applied three non-parametric statistical methods—reverse arrangement test, rank-sum test, and z-scores [9]—to both the basic and change features to select the critical ones. In this paper, we follow [17] to create and select *critical features*.

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

For “W”, the critical features are basic features 1–9 and 11 in Table III, along with the 6-hour differences of features 1, 9 and 11. For “S” and “M”, the critical features are 1–5, 8, and 10, and the 6-hour differences of features 1 and 3. We divide the datasets into training and test sets with respect to time. For each good disk, we take the earlier 70% of the samples as training data, and the later 30% as test data. Since the chronological order of disk failures was not recorded, we divide them randomly into training and test sets in a 7 to 3 ratio. Since good disks are far more numerous than failed disks, only some good samples are used to train the GBRT models. We randomly choose 3 samples for “W” and 1 sample for “S” and “M” per good disk in the training set as good samples to train GBRT models.

V. Experimental Results

A. Comparison of Metrics

In this section, we will illustrate how to evaluate disk failure prediction models (including binary classifiers and residual life predictions) using the new evaluation metrics (migration accuracy), vs.the previous evaluation metrics (prediction accuracy). We use the “W” dataset in these experiments.

1) Binary Classifier: In this experiment, we illustrate how we can evaluate binary classifiers using MR and MMR, along with FDR, FAR, and TIA.

The experiments in [17] and [20] show an advantage of the Classification Tree (CT) and Recurrent Neural Networks (RNN) models in predicting whether or not a disk is going to fail, over other binary classification models. We use these models and evaluate their performance in terms of MR, MMR, FDR, FAR, and TIA, adopting the practices in [17] and [20] to preprocess data and build the CT and RNN models, respectively.

For the CT model, we take the last 168 failed samples before the failure actually occurs from each failed disk in the training set (the time windows is set to 168 hours), to train the model. To detect failures, we use a naive detection algorithm: predicting a disk is going to fail if only one sample is classified as failed by the model. The CT model has the prediction performance of FDR = 95.49%, FAR = 0.09%, and TIA = 354.6 hours [17, Table IV].

For the RNN model, we also use the same naive detection algorithm to detect the test disks. The RNN model has the prediction performance of FDR = 98.47%, FAR = 0.5134%, and TIA = 294.0 hours.

When testing the models, we process the samples in the test set sequentially for each disk. If a disk is predicted to fail, we migrate the data from it at a certain rate for an hour, until all the data has been migrated or the disk fails, measuring the MR and MMR. For all failure predictions, the *migration transfer rate* is fixed, and set to one of $\{m/2, m/7, m/24, m/72, m/120, m/168, m/240, m/336\}$ TB/h, for a m TB disk.

TABLE IV: Performance of the CT and RNN on “W” dataset, in terms of MR and MMR, with different migration transfer rates. The storage capacity of a disk is m TB.

Rate (TB/h)	**CT**		**RNN**	
	MR (%)	**MMR (%)**	**MR (%)**	**MMR (%)**
FDR/FAR	95.49	0.09	98.47	0.5134
$m/2$	95.11	0.0900	98.47	0.4936
$m/7$	94.85	0.0634	98.36	0.4357
$m/24$	93.98	0.0302	96.31	0.3684
$m/72$	91.11	0.0126	92.49	0.177
$m/120$	89.83	0.0076	90.7	0.108
$m/168$	88.26	0.0054	88.1	0.0783
$m/240$	84.11	0.0038	81.57	0.0548
$m/336$	78.54	0.0027	73.79	0.0392

Table IV reports the migration accuracy of the CT and RNN models as binary classifiers, in terms of MR and MMR, with different migration transfer rates for pre-warning handling. We also include the FDR or FAR which is what would occur if all at-risk data were to be successfully migrated. With a migration transfer rate of $m/2$ TB/h for the CT model and $\geq m/7$ TB/h for the RNN model, MR is close to the FDR (i.e., data that was on a disk that was predicted to fail was always completely migrated). However, with such high migration transfer rates, the quality of service for users would drop, especially when simultaneous failure predictions are raised.

In practice, storage systems generally can not offer sufficient resources for successful migration for all migrations, only affording a relatively slow transfer rate. As such, there will be some at-risk data not migrated before failures occur despite being predicted by the models in advance, as in Table IV (MR deteriorates as the migration transfer rate decreases).

In addition, the RNN model has a better FDR than the CT model, which means the RNN method more accurately detects at-risk disks. However, when the transfer rate is $\leq m/168$ TB/h, the CT model has a higher MR, which means it protects more at-risk data. Compared with the previous evaluation metrics (FDR and FAR), the new metrics (MR and MMR) give users a more realistic measure of how a binary classifier will actually protect at-risk data.

When the voting-based detection algorithm in [17], [20] is used to test the models, there are the similar results as in Table IV.

2) Disk Residual Life Prediction: In this experiment, we illustrate how to compare disk residual life prediction models using the new evaluation metrics, MR and MMR, vs. the previous evaluation metric ACC.

The experiments in Xu et al. [20] also show the advantage of the RNN model in disk health status assessment over other models. So, when we evaluate our GBRT model, we use the RNN model (as disk residual life prediction model) as the control group, adopting the practices in [20] to preprocess data and build the RNN models.

When testing the models, we process the samples in the test set sequentially for each disk. If a disk’s residual life (the prediction result based on a sample) is mapped into level 6, where the pre-warning migration rate is specified as 0, we do nothing to the disk. If a disk’s residual life is mapped into one of the levels 1–5, we migrate the data from the disk at the specified rate for an hour, until all the data has been migrated, measuring the MR and MMR.

We also calculate the proportion $\mathrm{ACC_g}$ of good samples that are predicted at level 6, as the value of ACC for good disks, and calculate the proportion $\mathrm{ACC_f}$ of failed samples that are predicted at the right level (levels 1–5), as the value of ACC for failed disks.

Since the lengths of residual life intervals are unequal, each interval has a different number of training samples, which may have negative effects on the prediction performance of GBRT models. So, for each failed disk in training set, we do not use all the samples, but take out two samples evenly from every interval, to train the GBRT models. When we build the GBRT models, some important parameters are set as follows: learning rate $\alpha = 0.1$, number of iterations $c = 500$, and tree-depth $d = 4$. Unless otherwise stated, we use the same method for choosing failed training samples.

Table V reports the disk residual life prediction performance of the GBRT and RNN models, in terms of MR, MMR, $\mathrm{ACC_g}$, and $\mathrm{ACC_f}$.

For any of the three partitions, the RNN model has better performance in terms of $\mathrm{ACC_f}$ than the GBRT model, which means the RNN method more frequently predicts the residual life of failed samples in the right interval. However, the GBRT model has a higher MR, which means it protects more at-risk data. Crucially, prediction accuracy in terms of $\mathrm{ACC_f}$ or $\mathrm{ACC_g}$ gives misleading results: RNN significantly outperforms GBRT, despite successfully migrating (and thus protecting) less data.

The proposed metrics (MR and MMR), by measuring the amount of protected data, are more directly meaningful than $\mathrm{ACC_f}$ and $\mathrm{ACC_g}$. They thus offer a new means of evaluating disk residual life prediction models, particularly in the case of large systems with 10000+ disks.

Moreover, due to MR and MMR describe the actual target of disk failure prediction models, they can be used to compare the performance of binary classifiers and disk residual life prediction models, which can not be done using the traditional metrics.

B. Evaluating the Improved GBRT Algorithms

In Section III-C we suggest two improvements, (4) and (5), to the target residual calculation which could improve the disk residual life prediction. In this subsection, we test how effective the adjustments are. We use GBRT$*$ to denote when the first modification is used (where target residuals are set to 0 when samples are predicted in the correct intervals), and GBRT$**$ to denote when both modifications are used (where target residuals are set to 0 when samples are predicted in the tolerated intervals).

For this experiment, when training the GBRT models, we also adjust the residual life interval of good disks from ">500 hours" to ">800 hours" to reduce mismigration. However, when testing the models, we continue to use 500 hours as the boundary between good and failed. We continue to use the "W" dataset in this experiment.

The results are shown in Table VI. As expected, GBRT$*$ has better migration performance than the original GBRT model, while GBRT$**$ has the best performance. While these are small improvements, due to the large scale of cloud storage systems, even these small improvements in migration accuracy can be worthwhile.

TABLE VI: Performance of GBRT models on "W" dataset. The "GBRT$*$" and "GBRT$**$" respectively denote the improved GBRT models.
.

	Model	**MR (%)**	**MMR (%)**
Partition 1	GBRT	87.54	0.0028
	GBRT$*$	88.44	0.0012
	GBRT$**$	**88.72**	**0.0010**
Partition 2	GBRT	86.69	0.0017
	GBRT$*$	86.76	0.0021
	GBRT$**$	**88.42**	**0.0009**
Partition 3	GBRT	84.91	0.0021
	GBRT$*$	86.09	0.0007
	GBRT$**$	**86.50**	**0.0006**

In addition, when testing the models, we count the number of migration operations in each residual life interval, to evaluate the impact of pre-warning handling on system availability. Fig. 2 and 3 plot the distribution of migration transfer rates caused by the prediction results of GBRT$**$ and RNN models, using $partition1$, respectively.

For GBRT$**$, most migration operations for correct failure predictions are performed at relatively slow rates (no more than $m/72$ TB an hour), which have a minor impact on system availability. Almost all of the migrations for false failure predictions are performed at a very slow rate ($m/336$ TB an hour), which will have a negligible impact on system availability. This shows that the proposed GBRT$**$ model can protect almost 90% of at-risk data, while incurring only a minor reduction in availability. The migration rate distributions for GBRT and GBRT$*$ are similar to those in Fig. 2.

For RNN, we observe that migrations and mismigrations are performed at slower transfer rates (levels 3–5), which results in high ACC, but low MR. The RNN model also results in a large number of mismigrations vs. GBRT$**$ (consistent with Table V).

C. Simulating Practical Use

We evaluate the GBRT$**$ model by simulating its application in real-world data centers: being used with different disk families, being used in small-scale data centers, and being used with multiple disk models.

1) Performance with Different Disk Models: Different models of disks have different characteristics which may impact on their reliability, even if they are made by the same manufacturers. Consequently, effectiveness over varying disk models is an important factor in prediction models. To this end, we test the proposed GBRT$**$ model on the "S" and "M" datasets, which are composed of different disk models from that of the "W" dataset.

TABLE V: Performance of the GBRT and RNN on "W" dataset, in terms of MR, MMR, ACC_g, and ACC_f.

	Model	Previous Metrics		New Metrics	
		ACC_f (%)	ACC_g (%)	MR (%)	MMR (%)
Partition 1	RNN	30.28	99.333	78.54	0.0809
	GBRT	23.83	99.985	87.54	0.0028
Partition 2	RNN	27.02	99.846	77.85	0.056
	GBRT	22.31	99.987	86.69	0.0017
Partition 3	RNN	39.90	99.715	77.56	0.0415
	GBRT	19.90	99.984	84.91	0.0021

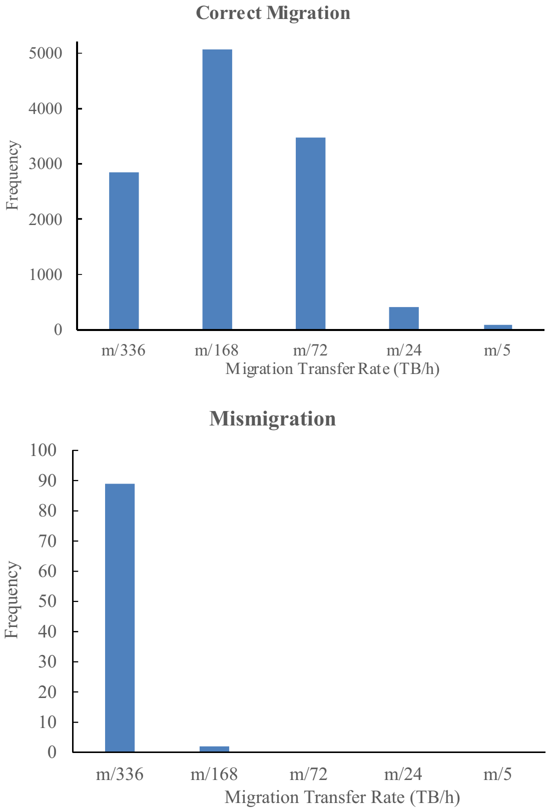

Fig. 2: Distribution of the frequencies of transfer rates for migration for the GBRT$**$ model. The storage capacity of a disk is m TB. There were no mismigrations at the $m/72$, $m/24$, nor $m/5$ transfer rates.

From the failed disks in training sets, to train the GBRT models, we take out three samples evenly from every residual life interval.

The results are shown in Table VII. On the "S" and "M" datasets, the GBRT$**$ model maintains as good performance in terms of MR and MMR, comparable with that on "W", which demonstrates the effectiveness of the proposed model as disk models vary.

2) Evaluating with Fewer Disks: The datasets used in above experiments, "W", "S" and "M", all involve a large number of disks, which are collected from two big data centers. In the real world, however, prediction models will often be used in small

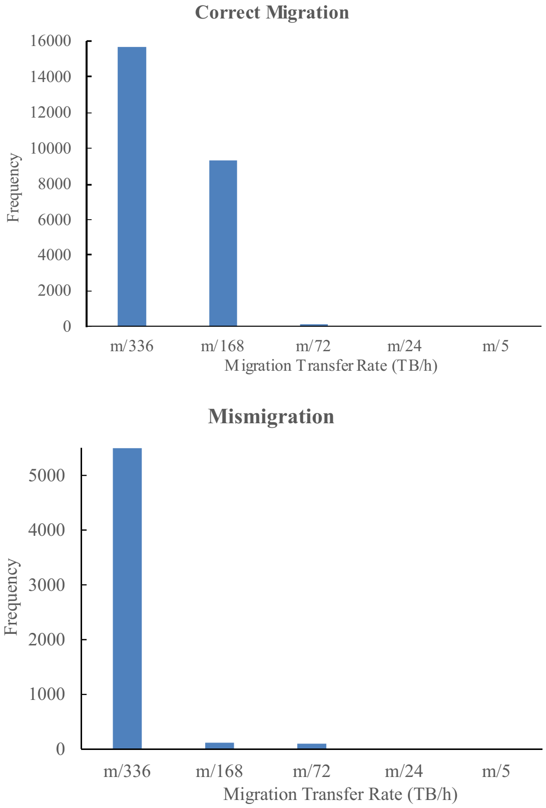

Fig. 3: Distribution of the frequencies of transfer rates for migration for the RNN model. The storage capacity of a disk is m TB. There were no migrations nor mismigrations at the $m/24$ and $m/5$ transfer rates.

TABLE VII: Migration performance of GBRT$**$ on the "S" and "M" datasets.

	Dataset	MR (%)	MMR (%)
Partition 1	"S"	95.58	0.0042
	"M"	93.87	0.0060
Partition 2	"S"	94.05	0.0041
	"M"	94.61	0.0084
Partition 3	"S"	92.56	0.0045
	"M"	91.87	0.0127

and medium-sized data centers. To evaluate the effectiveness of prediction model applying to small and medium-size data centers, we test it with synthesized datasets containing fewer disks. We create four small datasets, denoted W_1, W_2, W_3, and W_4, by randomly choosing 10%, 25%, 50%, and 75% of all the good and failed disks respectively from the "W" dataset. So the smallest dataset W_1 contains only 2,296 good disks and 43 failed disks.

Table VIII shows the prediction performance of the GBRT$**$ model with datasets $W_1, \ldots, W_4$. With all four datasets, GBRT$**$ obtains acceptable performance.

TABLE VIII: Migration performance of GBRT$**$ on small-sized synthesized datasets.

	Dataset	MR (%)	MMR (%)
Partition 1	W_1	88.86	0.0208
	W_2	91.42	0.0038
	W_3	94.71	0.0052
	W_4	87.41	0.0041
Partition 2	W_1	86.90	0.0150
	W_2	90.92	0.0043
	W_3	94.81	0.0034
	W_4	87.14	0.0038
Partition 3	W_1	82.28	0.0100
	W_2	89.54	0.0067
	W_3	93.49	0.0015
	W_4	86.24	0.0019

3) Performance with multiple disk models: It is not unusual for there to be multiple disk models in a real-world data center. Although building a distinct prediction model for every disk model is desirable, this would be an onerous task in practice (involving sampling over a long time period). Therefore, training prediction models using samples from different disk models will be necessary.

In order to simulate a single data center containing different disk models, we create a hybrid dataset (denoted "SM") by merging the "S" and "M" datasets, which were collected from a single data center. To train the GBRT$**$ model, for each failed disk in the training set, we take out three samples evenly from every residual life interval.

The migration performance of the GBRT$**$ model on the "SM" dataset is shown in Table IX, which is still acceptable for practical use.

TABLE IX: Performance of the GBRT$**$ on the "SM" dataset.

	MR (%)	MMR (%)
Partition 1	96.14	0.0079
Partition 2	96.45	0.0077
Partition 3	95.81	0.0066

4) Time cost: All the GBRT experiments are done on a standard PC desktop. The training of each GBRT model can be completed within 10 minutes, and the testing time is more less. Our proposed method is suitable for on-line running in large-scale storage systems.

VI. Conclusions

In this paper, we argue that the existing evaluation metrics (FDR, FAR, and ACC) for disk failure prediction models are insufficient for selecting and comparing models, particularly for large storage systems (such as cloud storage systems). We present two new metrics, MR and MMR, which directly measure how much at-risk data is actually protected and how much data is unncessarily protected, respectively.

In comparing two failure prediction models (the RNN model and the proposed GBRT model), we encounter the undesirable property where the RNN model makes better predictions (better ACC) but protects less at-risk data (worse MR) and unncessarily protects more data (worse MMR). Comparing these models only using ACC would therefore be misleading.

The GBRT model proposed in this paper predicts disks' residual life, allowing operators to migrate the at-risk data based on urgency, thereby ensuring both reliability and availability. We also propose a method for choosing suitable migration rates from the residual life predictions. Experimental results indicate that the GBRT model is suitable for practical use in real-world data centers.

VII. Acknowledgments

This work is partially supported by NSF of China (grant numbers: 61373018, 11301288, 11550110491), Program for New Century Excellent Talents in University (grant number: NCET130301) and the Fundamental Research Funds for the Central Universities (grant number: 65141021).

References

[1] G. Jung, M. A. Hiltunen, K. R. Joshi, R. K. Panta, and R. D. Schlichting, "Ostro: Scalable placement optimization of complex application topologies in large-scale data centers," in *Proc. ICDCS*, 2015, pp. 143–152.

[2] J. A. Aroca, A. Chatzipapas, A. F. Anta, and V. Mancuso, "A measurement-based characterization of the energy consumption in data center servers," *IEEE J. Sel. Area. Comm.*, vol. 33, no. 12, pp. 2863–2877, 2015.

[3] S. K. Rethinagiri, O. Palomar, A. Sobe, G. Yalcin, T. Knauth, J. R. T. Gil, P. Prieto, M. Schneegaß, A. Cristal, O. Unsal, P. Felber, C. Fetzer, and ragomir Milojevic, "ParaDIME: Parallel distributed infrastructure for minimization of energy for data centers," *Microprocess Microsyst.*, vol. 39, no. 8, pp. 1174–1189, 2015.

[4] T. Wang, Z. Su, Y. Xia, J. K. Muppala, and M. Hamdi, "Designing efficient high performance server-centric data center network architecture," *Computer Networks*, vol. 79, pp. 283–296, 2015.

[5] Q. Xin, E. L. Miller, and S. T. J. Schwarz, "Evaluation of distributed recovery in large-scale storage systems," in *Proc. IEEE HPDC*, 2004, pp. 172–181.

[6] S. Mitra, R. K. Panta, M.-R. Ra, and S. Bagchi, "Partial-parallel-repair (PPR): a distributed technique for repairing erasure coded storage," in *Proc. EuroSys*, 2016, pp. 1–14.

[7] V. Gramoli, G. Jourjon, and O. Mehani, "Disaster-tolerant storage with sdn," in *Proc. NETYS*, 2015, pp. 278–292.

[8] B. Allen, "Monitoring hard disks with SMART," *Linux Journal*, no. 117, 2004.

[9] J. F. Murray, G. F. Hughes, and K. Kreutz-Delgado, "Machine learning methods for predicting failures in hard drives: A multiple-instance application," *J. Mach. Learn. Res.*, vol. 6, pp. 783–816, 2005.

[10] G. Hamerly and C. Elkan, "Bayesian approaches to failure prediction for disk drives," in *Proc. Conference on Machine Learning*, 2001, pp. 202–209.

[11] G. F. Hughes, J. F. Murray, K. Kreutz-Delgado, and C. Elkan, "Improved disk-drive failure warnings," *IEEE Trans. Reliability*, vol. 51, no. 3, pp. 350–357, 2002.

[12] J. F. Murray, G. F. Hughes, and K. Kreutz-Delgado, “Hard drive failure prediction using non-parametric statistical methods,” in *Proc. Artificial Neural Networks*, 2003.

[13] Y. Zhao, X. Liu, S. Gan, and W. Zheng, “Predicting disk failures with HMM- and HSMM-based approaches,” in *Proc. Advances in Data Mining: Applications and Theoretical Aspects*, 2010, pp. 390–404.

[14] B. Zhu, G. Wang, X. Liu, D. Hu, S. Lin, and J. Ma, “Proactive drive failure prediction for large scale storage systems,” in *Proc. MSST*, 2013, pp. 1–5.

[15] Y. Wang, Q. Miao, and M. Pecht, “Health monitoring of hard disk drive based on Mahalanobis distance,” in *Proc. Prognostics and System Health Management Conference*, 2011, pp. 1–8.

[16] Y. Wang, Q. Miao, E. W. M. Ma, K.-L. Tsui, and M. G. Pecht, “Online anomaly detection for hard disk drives based on Mahalanobis distance,” *IEEE Trans. Reliability*, vol. 62, no. 1, pp. 136–145, 2013.

[17] J. Li, X. Ji, Y. Jia, B. Zhu, G. Wang, Z. Li, and X. Liu, “Hard drive failure prediction using classification and regression trees,” in *Proc. Dependable Systems and Networks*, 2014, pp. 383–394.

[18] A. Ma, F. Douglis, G. Lu, D. Sawyer, S. Chandra, and W. Hsu, “RAIDShield: characterizing, monitoring, and proactively protecting against disk failures,” in *Proc. USENIX FAST*, 2015, pp. 241–256.

[19] S. Wu, H. Jiang, and B. Mao, “Proactive data migration for improved storage availability in large-scale data centers,” *IEEE Trans. Comput.*, vol. 64, no. 8, pp. 2637–2651, 2015.

[20] C. Xu, G. Wang, Z. Li, and X. Liu, “Health status and failure prediction for hard drives with recurrent neural networks,” accepted for publication in IEEE Trans. Comput.

[21] S. Pang, Y. Jia, R. J. Stones, X. Liu, and G. Wang, “A combined Bayesian network method for predicting drive failure times from S-MART attributes,” accepted for publication in IJCNN 2016.

[22] S. Wu, H. Jiang, and B. Mao, “Proactive data migration for improved storage availability in large-scale data centers,” *Computers, IEEE Transactions on*, vol. 64, no. 9, pp. 2637–2651, 2015.

[23] A. Qin, D. Hu, J. Liu, W. Yang, and D. Tan, “Fatman: Cost-saving and reliable archival storage based on volunteer resources,” *Proc. VLDB Endowment*, vol. 7, no. 13, pp. 1748–1753, 2014.

[24] X. Ji, Y. Ma, R. Ma, P. Li, J. Ma, G. Wang, X. Liu, and Z. Li, “A proactive fault tolerance scheme for large scale storage systems,” in *Algorithms and Architectures for Parallel Processing*. Springer, 2015, pp. 337–350.

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

[26] J. H. Friedman, “Greedy function approximation: a gradient boosting machine,” *Annals of Statistics*, pp. 1189–1232, 2001.

[27] B. Schroeder and G. A. Gibson, “Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you?” in *Proc. FAST*, 2007.