Health Status Assessment and Failure Prediction for Hard Drives with Recurrent Neural Networks

Chang Xu, Gang Wang*, Xiaoguang Liu*, Dongdong Guo, and Tie-Yan Liu, *Senior Member, IEEE*

Abstract—Recently, in order to improve reactive fault tolerance techniques in large scale storage systems, researchers have proposed various statistical and machine learning methods based on SMART attributes. Most of these studies have focused on predicting failures of hard drives, i.e., labeling the status of a hard drive as "good" or not. However, in real-world storage systems, hard drives often deteriorate gradually rather than suddenly. Correspondingly, their SMART attributes change continuously towards failure. Inspired by this observation, we introduce a novel method based on Recurrent Neural Networks (RNN) to assess the health statuses of hard drives based on the gradually changing sequential SMART attributes. Compared to a simple failure prediction method, a health status assessment is more valuable in practice because it enables technicians to schedule the recovery of different hard drives according to the level of urgency. Experiments on real-world datasets for disks of different brands and scales demonstrate that our proposed method can not only achieve a reasonable accurate health status assessment, but also achieve better failure prediction performance than previous work.

Index Terms—Hard drive failure prediction, SMART, Health degree, Recurrent Neural Networks.

1 Introduction

In this cloud computing and big data era, the reliability of cloud storage systems (data centers) is a major challenge that IT enterprises have to face. According to [1], [2], the hard drive is one of the main sources of failure in today's data centers. Although the theoretical annual failure rate of a single hard drive could be lower than 1%, the real annual failure rate observed in data centers could exceed 10% [2]. It was estimated in [3] that in a petabyte-level file system, hard drives fail almost every day—the large scale of a data center magnifies the failure probability of hard drives, making hard drive failures the norm rather than an exception.

In response to the problem of hard drive failure, researchers have investigated on both reactive fault tolerance and proactive failure prediction. Different from reactive fault tolerance (e.g., designing erasure codes to improve storage system reliability), proactive failure prediction forecasts hard drive failures before they actually happen, and therefore can inform technicians to take actions in advance. To improve the accuracy of proactive failure prediction, in recent years, statistical and machine learning methods have been adopted to build prediction models based on the SMART (Self-Monitoring, Analysis and Reporting Technology) attributes [4], [5], [6], [7], [8], [9], [10], [11]. Although these methods have demonstrated their effectiveness in a number of circumstances, they have clear limitations. For example, these prediction models only yield binary classification on the status of a hard drive (i.e., good or bad), and cannot distinguish between being close to failure and still being far from failure. As another example, most of these methods take a single snapshot of the SMART attributes as the input instance for prediction, without considering the dependency between different statuses of a hard drive in the time horizon. These limitations motivate us to explicitly model sequential information using SMART attributes so as to gauge the different health statuses of hard drives.

In real-world storage systems, SMART attributes (e.g., *Seek Error Rate* and *Power On Hours*) are logged with time stamps, in order to monitor internal attributes of individual hard drives and to raise alarms if any attribute exceeds its threshold. A hard drive often deteriorates gradually, rather than abruptly. Correspondingly, the SMART attributes change continuously towards the status of failure. Thus, it is natural to employ temporal analysis methods to model the sequential dependency between SMART attributes over time. Recurrent Neural Networks (RNN) have been proven an effective tool to model temporal dependency in various applications, such as language models [12], [13], speech recognition [14], machine translation [15], and so on. This inspires us to consider leveraging RNN in the assessment of the health status and the prediction of failure for hard drives.

Different from traditional feedforward neural networks, RNNs can exploit their internal memory to analyze the temporal sequences of inputs. There are basically two kinds of temporal dependencies: short-term dependency (such as the Markovian properties) and long-range dependency (such as those in natural languages). An RNN is especially effective in modeling long-range dependencies. In this work, we show that the health status of hard drives also has long-range dependency, therefore it is natural and appropriate to

- *C. Xu, G. Wang, and X. Liu are with the Department of Computer Science and Information Security, Nankai University, Tianjin, China, 300071. E-mail: {changxu, wgzwp, liuxg}@nbjl.nankai.edu.cn*
- *D. Guo is with Qihoo 360 Technology Company, Beijing, China, 10001. E-mail: guodongdong@360.cn*
- *T. Liu is with Microsoft Research Asia, Beijing, China, 100010. E-mail: tyliu@microsoft.com*

leverage RNN to assess the health status and predict failures of hard drives via the sequence of SMART attributes. Specifically, in the learning process, we feed the SMART attributes in each time point to the hidden layer of RNN, together with its previously accumulated hidden states. In this way, the dependency among instances will be embedded into the RNN structure. In addition, we adopt a discrete classification method to define the levels of health status (i.e., health degree). The discrete classification can indicate the remaining life of hard drives, and can be used to raise alarms if needed. Our experiments on real-world datasets reveal that our proposed RNN-based method can not only achieve reasonable accuracy on health status assessment, but also achieve better failure prediction performance than previous work.

The remaining part of the paper is organized as follows. First, we survey the existing work on hard drive failure prediction based on SMART attributes in Section 2. Then, we propose our RNN-based model in Section 3. After that, we present our experimental results in Section 4. We conclude the paper in Section 5.

2 Related Work

Compared with the traditional passive fault tolerance technique, proactive drive failure prediction tends to provide more opportunities to handle potential failures in advance, and thereby greatly reduce negative impacts on system reliability and availability when failure occurs. Accordingly, a lot of research has been done especially on the topic of SMART based proactive fault tolerance technique.

SMART is a monitoring system that detects and reports on various indicators of drive reliability and it is widely used for hard drive failure prediction. Threshold-based algorithms are used to predict the drive failure, but the performance is far from satisfying, in that **FDR** (failure detection rate, the fraction of failed drives that are correctly classified as failed) is 3-10%, and **FAR** (false alarm rate, the fraction of good drives that are incorrectly classified as failed) is 0.1%. There is still a long way to go before applying the disk drive failure prediction technology to practice.

Many learning-based methods have been proposed to improve the performance of drive failure prediction based on SMART records data. Hamerly and Elkan [5] proposed two Bayesian approaches to predict hard drive failures on a small dataset (containing 1,927 disk drives in total, but only 9 drives which fail) collected from Quantum Inc. One of the methods they used was named as NBEM and the other one was a naive Bayes classifier. Under 1% FAR, NBEM achieved 35-40% prediction accuracy, and the naive Bayes classifier achieved 55%.

Hughes et al. [6] proved that most of the meaningful SMART attributes are non-parametrically distributed. Inspired by this observation, they used the Wilcoxon rank-sum test for predicting hard drive failure. They proposed two different strategies: a multivariate test and an OR-ed single attribute test. The highest FDR achieved by these two methods was 60% at 0.5% FAR on a small dataset of 3,744 drives.

Murray et al. [7] compared the performance of four different methods including SVM, unsupervised clustering, rank-sum and reverse arrangements test. The results showed that the rank-sum method achieved the best performance (33.2% FDR at 0.5% FAR). They also proposed a new algorithm named mi-NB (Multiple-Instance Naive Bayes) [4]. The results showed that ranksum test outperformed SVM for a certain small set of SMART attributes (28.1% FDR at 0% FAR). When using all features, SVM achieved the best performance (50.6% detection rate with 0% FAR). Note that all of the methods compared in their work were evaluated on a small dataset containing only 369 disk drives (good and failed drives are about half and half), which is also used in several other publications. But this dataset does not match the situation in real-world data centers. Moreover, since it was collected before 2003, the SMART information format is not consistent with the current SMART standard. These factors undermine the practicability of models.

Wang et al. [10] proposed a method for drive anomaly prediction based on Mahalanobis distance. The experimental results on the dataset used in [5] showed that the method with prioritized attributes selected by FMMEA (Failure Modes, Mechanisms and Effects Analysis) performed better than the one with all attributes. In their later work [11], by using minimum redundancy maximum relevance, the redundant attributes were filtered out from the attributes set selected by FMMEA. They then built up a baseline Mahalanobis space using the good drive data of the critical parameters. This approach could detect nearly 67% of the failed drives at 0% FAR, and most of the failed drives could be detected about 20 hours in advance.

Recently, Backpropagation Artificial Neural Network [9] and Classification Trees [16] have been shown to achieve great improvement on predicting drive failures based on SMART attributes. A real-world dataset containing 23,395 drives was used in these papers. The BP ANN model could reach an excellent FDR which was up to 95% with a reasonable low FAR. While the Classification and Regression Tree models perform better in prediction performance as well as stability and interpretability.

The aforementioned methods all take every SMART sample as an input instant, but ignore the time-sequence information of SMART attributes which can reflect trends in the changing health status of drives. Zhao et al. [8] applied Hidden Markov Models and Hidden Semi-Markov Models to predict hard drive failures. They used the time-sequence information of SMART attributes. By using the best single attribute, the HMM and HSMM models had an FDR of 46% and 30%, respectively. Even by combining the best two attributes, the HMM model only reached a FDR of 52%. Although their models outperformed many other methods which paid no attention to the relationship of attribute values over time, their performance is still far below the state of the art.

Meanwhile, a few most recent studies leveraging Recurrent Neural Networks (RNN) for modeling the long-term dependent sequential data have achieved great success. For example, the RNN language model [12], [13] successfully leveraged long-span sequential information within a massive language corpus, which results in better performance than the traditional neural networks language model [17]. Moreover, RNN-based handwriting recognition [18], speech recognition [14], and machine translation [15] systems have also led to much improvement in their corresponding tasks. Compared to traditional feedforward neural networks and other short-term dependency models, RNN has demonstrated its strong capability to exploit sequential data due to its specific recurrent network structure.

In contrast to all the aforementioned works, in this paper, we present a RNN-based method to leverage sequential information for predicting hard drive failures. Aiming at monitoring the health status of hard drives, we also adopt a multi-level classification in the output layer of the neural network. As a result, our new model is able to both predict failures and give drive health statuses using sequential SMART information, and is more accurate in predicting and more useful in practice than previous works.

3 The Proposed Method

We argue that the hard drive failure prediction problem belongs to long-range dependency in Section 4.2, so an RNN-based model with ability of modeling long-range dependent sequences is natural for this task. In this section, we introduce the RNN-based models.

3.1 Model

Artificial neural networks (ANNs) are a family of models inspired by biological neural networks and are generally presented as interconnected "neurons". The connections have numeric weights that can be tuned based on experience, which make them capable of learning. RNNs are a class of artificial neural networks where connections between neurons form a directed cycle, which allows it model temporal behaviors.

In our work, we use an architecture that is usually called a simple recurrent neural network as Figure 1 shows. This architecture is very easy to implement and train. It consists of an input layer i, an output layer y, a hidden layer h with recurrent connections, plus the corresponding weight matrices. Input to the network in time t is vector $i(t)$ that represents the features of SMART attributes at time t. We use $h(t)$ to denote the output of the hidden layer in time t, which also maintains a representation of the history of SMART attributes. The recurrent connections R between $h(t-1)$ and $h(t)$ can propagate sequential signals, where the vector $h(t-1)$ represents the values in the hidden layer computed from the previous step. The activation values of the hidden and output layers are computed as:

$$h(t) = f\big(U\, i(t) + R\, h(t-1)\big),$$
$$y(t) = g\big(V\, h(t)\big),$$

where $f(z)$ and $g(z)$ are sigmoid and softmax activation functions (the softmax function in the output layer is used to ensure that the outputs form a valid probability distribution, i.e., all outputs are greater than 0 and their sum is 1):

$$f(z) = \frac{1}{1+e^{-z}},$$
$$g(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}}.$$

The hidden layer can be considered as an internal memory which records dynamic sequential states. The recurrent

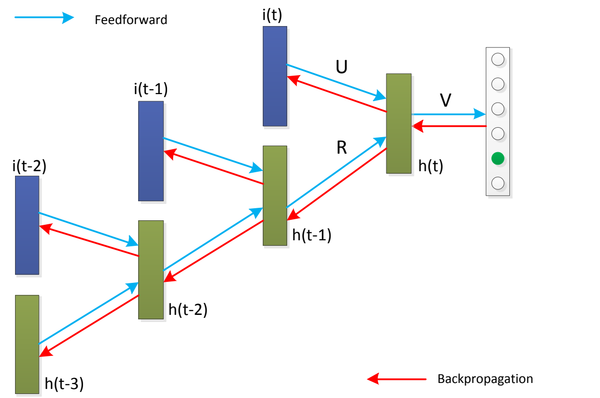


Fig. 1. RNN training process with the BPTT algorithm. Unfolding step is set to 3 in this figure.

structure is able to capture the historical context of health statuses. This makes RNN suitable for the tasks related to sequential prediction.

In our method, $i(t)$ represents the SMART attributes and $h(t)$ represents sequential information of a hard drive's previous health status. Thus, our prediction depends not only on the current input features, but also on the sequential historical information. The vector in the output layer $y(t)$ represents the health degree probability distribution.

3.2 Health Degree

As observed in real-world data centers, before a hard drive fails completely, there is a gradual trend towards abnormal status in SMART attributes. That is, there is a gradual process of deterioration in health status. In this paper, we quantify the health status of a hard drive as the time before it breaks down, and we define a drive's health degree by dividing the remaining time into different ranges according to the time before failure. For example, if the remaining time is very short, this means its health status is quite poor, and it is urgent for the technician to handle this failure alert.

Compared with the traditional binary failure prediction methods, health status prediction can significantly improve the reliability and availability of large scale distributed storage systems. Technicians or warning handlers could schedule the recovery of different hard drives and allocate system resources according to failure urgency and the remaining life time, so that we can balance the quality of user services and data migration. In a multiple failure situation, migration priority can be decided according to the health status of the drives. As a result, the probability of missing the most urgent failures decreases, and the relevant economic loss would also be avoided.

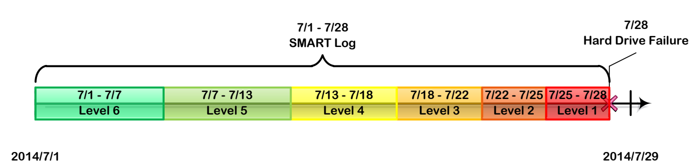


Fig. 2. An example of health degree settings. The health status of hard drive is splitted into 6 health degree levels. The closer to the time point when the hard drive break down, the lower the health degree is.

Figure 2 gives a possible instance of health degree settings where the health degree is divided into 6 classes according to the remaining time. Level 6 indicates that the disk drive works properly. Level 5 represents that the health status of the disk drive is fair. Levels 1-4 means that the disk drive is going to fail. In particular, Level 1 is the "red alert" which means that the remaining time is less than 72 hours for the current hard drive, so the alert must be dealt with immediately. Note that the time intervals for different health degree levels could be set differently, and the time intervals could be more sparse for prediction times far away from the fault time because of the low urgency.

We use this health degree setting in all of the experiments in this paper. Since we just choose this setting intuitively and we don't compare this setting with any other health degree setting, it might be possible to get better results with other carefully selected health degree settings.

3.3 Training

In the objective function, we aim to maximize the likelihood of correct prediction:

$$f(\lambda) = \sum_{t=1}^{N} \log y_{l_t}(t), \tag{1}$$

where the training samples are labeled $t = 1 \ldots N$ and l_t is the index of the correctly assessed health degree for the t-th sample.

The classical backpropagation for feedforward neural network training does not leverage some potential useful information such as the previous n samples of the training data, which is also related to the failure prediction and health degree assessment. Thus, in this work, we use another standard training method called "Backpropagation Through Time" or BPTT, which is a generalization of backpropagation for feedforward networks. BPTT was proposed in [19], and has been used in many practical applications, such as the RNN language model [12], [14]. Although BPTT may lead to local optimal values, it is much more efficient than other global optimization methods, especially on large data set. Actually, for common training sets and reasonable choices of neural network architecture and parameters, BPTT often efficiently finds a local optimum of the objective function that is good enough for practical purposes. Since we aims to deal with the large-scale training data and to perform on-line learning, BPTT is a good choice.

We apply the BPTT algorithm to the RNN based health degree assessment models as shown in Figure 1. The overall training pipeline can be unfolded into a deep neural network with n hidden layers, where the recurrent weight matrices R are identical and shared among these hidden layers. In this approach, the hidden layer of RNN can actually exploit the information of the most recent inputs and put more importance to the latest input, which matches the notion of sequential dependency.

The network is trained using Stochastic Gradient Descent (SGD). The gradient of the output layer is computed as

$$e_o(t) = d(t) - y(t),$$

where $y(t)$ is the assessed health degree probability, and $d(t)$ is the target 1-of-v vector indicating the health degree that it belongs to. The weights V between the hidden layer $h(t)$ and output unit $y(t)$ are updated as

$$V(t+1) = V(t) + h(t)e_o(t)^T\alpha - V(t)\beta,$$

where α is the learning rate, β is L2 regularization parameter, and $e_o(t)^T$ is the transposition of $e_o(t)$. Then, gradients of errors are propagated from the output layer to the hidden layer as

$$e_h(t) = d_h(e_o(t)^T V, t),$$

where the error vector is obtained using the function d_h that is applied element-wise

$$d_{hj}(x,t) = x h_j(t)(1 - h_j(t)).$$

The weight matrices U between the input layer $i(t)$ and the hidden layer $h(t)$ are then updated as

$$U(t+1) = U(t) + i(t)e_h(t)^T\alpha - U(t)\beta.$$

The recurrent weight matrices R are updated as

$$R(t+1) = R(t) + h(t-1)e_h(t)^T\alpha - R(t)\beta.$$

3.4 Inference

We illustrate the testing process of the RNN based health degree assessment model in Figure 3. The test data is also organized as ordered SMART attributes sequences. We feed forward the current SMART sample, together with the outputs of the hidden state of the previous SMART sample to get the current hidden state. Then the model makes the assessment.

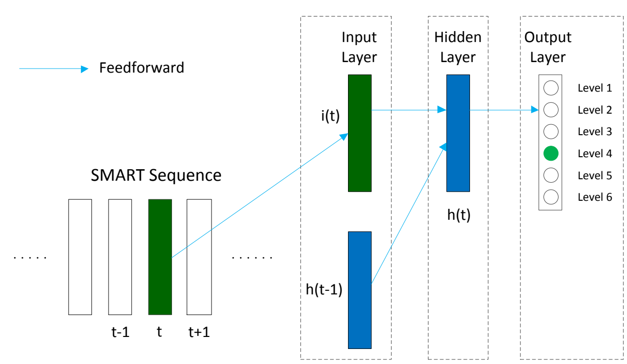


Fig. 3. The testing process of the RNN-based health degree assessment model. The inputs are the sequential SMART records. The outputs of the hidden state are computed based on the $(t-1)$-th SMART sample and are used as the inputs, together with the t-th SMART sample, to assess the health degree of disk drive at time t.

4 Experimental Results

In this section, we justify what kind of dependency the health status assessment and failure prediction has, and test different aspects of the performance of our proposed method in a series of experiments.

TABLE 1
Dataset details.

Brand	# of Good Drives	# of Drives which Fail
"W"	22790	434
"S"	38819	170
"M"	10010	147

4.1 Dataset

We used a dataset collected from a real-word data center which was released in [9] in our experiment. The brand and environment are two key factors which can affect the reliability of drives, so we collect two more datasets from another real-word data center to evaluate our proposed method and other methods. These three datasets are represented as "W", "S" and "M" according to their brand. All of the drives in these datasets were labeled to be either good or not. Table 1 lists the details of the datasets.

We use these three datasets to evaluate our proposed method and other models. Each dataset is divided into training and test sets. For each good drive, we take the earlier 70% of the samples within a week as training data, and the later 30% as test data. Since failed drives are much less common than good drives, we used all failed drives and randomly divide them into training and test sets in a 7 to 3 ratio.

We follow the previous work [4], which used three non-parametric methods: reverse arrangement test, rank-sum test and z-scores to select features from 23 meaningful attributes in SMART records. Afterwards, there were 10 attributes left, as shown in Table 2. Each SMART attribute has a six-byte width raw value which is vendor-specific and a normalized value ranging from 1 to 253. Since some normalized values lose precision and their corresponding raw values are more sensitive to the health condition of drives, we select the raw values of the 3rd and 5th attributes in addition to the normalized values of other attributes in Table 2 to build our models. The 6-hour change rates of three attributes (the 1st, 3rd and 4th attributes in Table 2) were also selected. To verify the effectiveness of the selected features, we apply our proposed RNN model to several different feature sets. Similar to the experimental results in [16], the 13 selected features outperform other feature sets in hard drive failure prediction task on dataset "W". Since some attribute values were not recorded for datasets "S" and "M", only the 1st, 3rd, 5th, 6th, 8th, 10th and 6-hour change rates of 1st and 3rd attributes were selected for training and testing on datasets "S" and "M". We rescale the range of all selected features to $[0, 1]$ by

$$x_{\text{normal}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}$$

where x is the original value of a feature, and x_{max} and x_{min} are the maximum and minimum values of this feature in training set, respectively. Also, x_{normal} is set to 1 if x in test set is larger than x_{max}, and it is set to 0 if x in test set is smaller than x_{min}.

4.2 Dependency Analysis

For hard disk health status assessment and failure prediction, although the health status of disks change gradually and mostly monotonically, the SMART attributes are not stable. For instance, "Temperature Celsius" may change significantly if the disk is frequently read or written to within a short period. To avoid this kind of change of attributes confusing the drive's health assessment, the temperature of the disk over a long time period should be considered.

TABLE 2
Selected attributes in SMART records.

ID	Attributes	ID	Attributes
1	Raw Read Error Rate	6	Spin Up Time
2	Reported Uncorrectable Errors	7	High Fly Writes
3	Reallocated Sectors Count	8	Temperature Celsius
4	Hardware ECC Recovered	9	Seek Error Rate
5	Current Pending Sector Count	10	Power On Hours

The long-range dependency can be regarded as a kind of high-order Markov property: the current state is highly correlated with earlier states. In this paper, we measured the Markov dependency by comparing the Conditional Entropy of different order feature representations. Let time sequence data $\{a^1, a^2, \ldots, a^n\}$ denote the features from time point 1 to time n. The Conditional Entropy can be defined as:

$$H(Y|X) = -\sum_{x \in X} p(x) \sum_{j=0}^{m} p(y_j|x) \log p(y_j|x) \tag{2}$$

where y is the health degree, m is the number of health degree labels ($m = 6$ in this paper). For order-1 feature representations, $x = a^t$ (the features at time t). For order-n feature representations, $x = a^t a^{t-1} \cdots a^{t-n+1}$ (merging the features from time t to $t - n + 1$). We keep one decimal place for all float features. Figure 4 shows the Conditional Entropy for varying order feature representations on dataset "W".

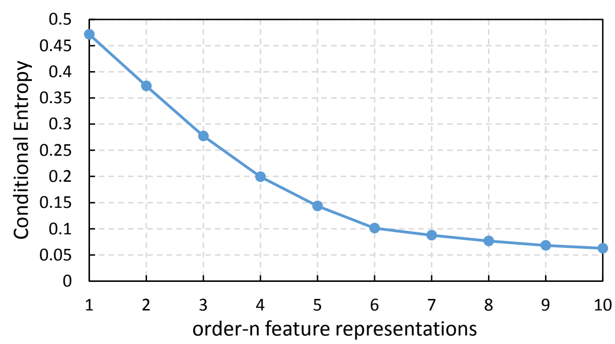


Fig. 4. Conditional Entropy for different order feature representations.

We see that the conditional entropy significantly decreases from order-1 to order-6. This task should belong to long-range dependency.

4.3 Compared Methods

The experiments in this paper focus on two tasks: one is failure prediction and the other is health degree assessment. For failure prediction, we compare RNN with other five methods as listed below.

- **HMM** Following the method of [8], we separately train two Hidden Markov Models with a mixture of Gaussian outputs for health disk drives (Positive model) and failure disk drives (Negative model) based on our selected features. A test drive would be predicted to fail if the difference between the sequence log-likelihood observed from positive model and negative model are both greater than the threshold. Otherwise, it is predicted as healthy.

- **Binary NN** Following the method of [9], Binary NN is an artificial neural network whose output layer only contains one node. A drive would be predicted to fail only if the output of the neural network is smaller than a threshold. Otherwise, the drive would be predicted to be healthy.
- **CT** Following the method based on classification tree [16]. To build the classification tree, information gain is used as splitting function. And the outputs of CT are also binary, which make it only able to predict whether the drive is healthy or not.
- **Multiclass NN** The architecture of Multiclass NN is similar to Binary NN, except its output layer, which contains 6 nodes, can be used to assess the health degree.
- **CRF** We also apply Conditional Random Fields (CRFs) with MIRA training for multiclass health degree assessment.

The outputs of HMM and CT are binary, which only predicts whether the drive is healthy or not rather than the health degree of it. Although the probabilistic output of Binary NN can be used to distinguish the degree of failure, it is not natural to assess the health degree according to a probabilistic output of a binary classification model, because it gives no information about how much time is left before the disk breaks down. Thus we only compare Multiclass NN, CRF and RNN for the health degree assessment task.

4.4 Experimental Setup

For CT and Binary NN, we follow the settings of [16] and [9], respectively, and we obtain similar results. For HMM, we also follow the experimental setting of [8]; several HMMs with the number of states varying from 10 to 50 were trained, and the one that maximizes the sequence log-likelihoods of training sequences was selected. For CRF, we use the selected features as unigram features along with the combination of the previous output health degree and current attributes as bigram features for training. Note that we fix the configurations and parameters of RNN rather than fine tune them on different datasets in our experiments. Specifically, the coefficient of weight decay is set to 10^{-7}; the learning rate is initially set to 0.1, and we divide the learning rate by 2 every 100 training epochs; the number of training epochs is 2000; and the size of hidden layer is 10. Instead of using all of the healthy drives for training [9], [16], we randomly select a portion of the healthy drives for training HMM, CRF, RNN and Multiclass NN, ensuring that the number of healthy drives for training is 10 times the number of failed drives. We perform all the experiments on a standard PC desktop since none of these methods require significant computer resources. The training of every method compared in this paper takes under 10 minutes, and the speed of failure/health degree prediction is nearly 10,000 disks per second. The time cost of our proposed method is suitable for on-line real-time monitoring of large-scale data centers.

SMART attributes of a hard drive within a short time interval are often very similar, in which case they are mapped to the same health degree. In order to leverage the relatively long sequence of historical information to assess the current health degree, we only sample one SMART record in each 24-hour period for training sequence dependency models HMM, CRF and RNN. As for the test data, we split the SMART records into 24 groups $\{D_1, D_2, \ldots, D_{24}\}$, where D_i is the collection of SMART records of the i-th hour in each 24-hour period. Then we separately test these 24 groups of data by using the trained model, and merge the 24 sets of prediction results together according to time stamps.

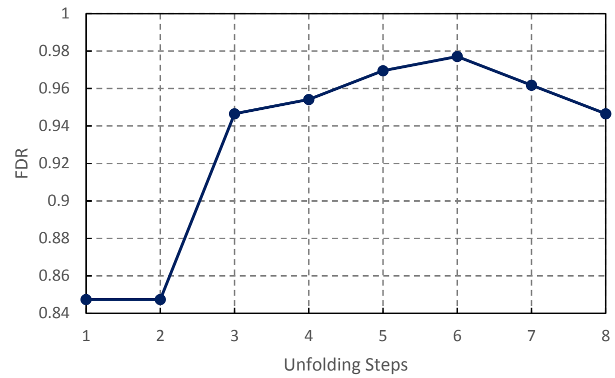


Fig. 5. Failure detection rate for RNN on dataset "W" as the number of unfolding steps varies.

As we described in Section 3, the unfolding structure plays an important role in modeling sequential dependency. Since the number of unfolding steps can directly determine the depth of sample sequence modeling, we delve into the performance of failure detection as the number of unfolding steps varies. According to our experimental results in Figure 5, FDR initially increases with an increasing number of unfolding steps. The best FDR is attained when unfolding 6 steps for RNN, after which the performance drops down. This conclusion is consistent with the dependency analysis, that the order-6 Markov Model is more suitable for this task. Therefore, we set the number of unfolding steps of RNN to 6 for the datasets in this paper. By checking the error terms during the BPTT process, we discover that the backpropagated error vanishes after 6 steps of unfolding, which explains why a larger number of unfolding step is detrimental.

We apply the voting-based failure detection algorithm [9] to evaluate the hard disk failure prediction performance of CT and Binary NN. Given the last N consecutive samples before a time point, the voting-based failure detection algorithm predicts that a drive is going to break down if more than $N/2$ samples are classified as failed, otherwise the drive will be classified as healthy. For Multi-class NN, CRF and RNN, we interpret the outputs with health degree level 1-4 as failed, and level 6 as healthy. Since the fair health status corresponding to level 5 is regarded as a intermediate state between healthy and failed, we don't use it for voting.

Here we propose two new voting-based failure detection algorithms for Multiclass NN, CRF and RNN. Given the last N consecutive samples before a time point, the failure detection rate follows two different criteria:

- VAT2H (Voting Algorithm which Tends to Health):

$$L^d = \begin{cases} \text{Healthy}, & \text{if } \sum_{i=1}^{4} C_i^d \geq C_6^d \\ \text{Failure}, & \text{if } \sum_{i=1}^{4} C_i^d < C_6^d \end{cases}.$$

TABLE 3
Overall performance of different models in terms of FDR and FAR.

Methods	"W"		"S"		"M"	
	FDR (%)	FAR (%)	FDR (%)	FAR (%)	FDR (%)	FAR (%)
HMM	57.69	0.34	94.12	0.38	75.56	1.02
Binary NN	84.21	0.07	94.12	0.08	95.56	0.84
CT	93.23	0.01	**96.08**	0.45	95.56	0.68
Multiclass NN (VAT2F)	83.21	0.70	92.16	0.10	95.56	0.60
Multiclass NN (VAT2H)	83.21	0.60	92.16	0.09	93.33	0.34
CRF (VAT2F)	85.50	0.23	92.16	0.09	68.88	0.10
CRF (VAT2H)	85.50	0.22	92.16	0.08	60.00	0.04
RNN (VAT2F)	**97.71**	0.06	**96.08**	0.05	**97.78**	0.59
RNN (VAT2H)	87.79	**0.004**	**96.08**	**0.04**	**97.78**	**0.03**

- VAT2F (Voting Algorithm which Tends to Failure):

$$L^d = \begin{cases} \text{Healthy}, & \text{if } \sum_{i=1}^{4} C_i^d > C_6^d \\ \text{Failure}, & \text{if } \sum_{i=1}^{4} C_i^d \le C_6^d \end{cases}.$$

where $\sum_{i=1}^{6} C_i = N$, and L_d is the failure prediction results for drive d, and C_i^d is the number of samples which are predicted as health level i for disk d. Health level 5 abstains from voting in the two failure detection criteria above.

4.5 Failure Prediction for Hard Drives

Figure 6 plots the prediction results of Multiclass NN, CRF and RNN using the two different voting-based failure detection algorithms on dataset "W". As expected, VAT2H leads to lower FAR than VAT2F in that more drives tend to be predicted as healthy, while VAT2F leads to higher FDR than VAT2H. When 47 voters are used, RNN and Multiclass NN both achieved their best performance. The FDR of RNN is higher than that of the other two models using either of the two voting algorithms.

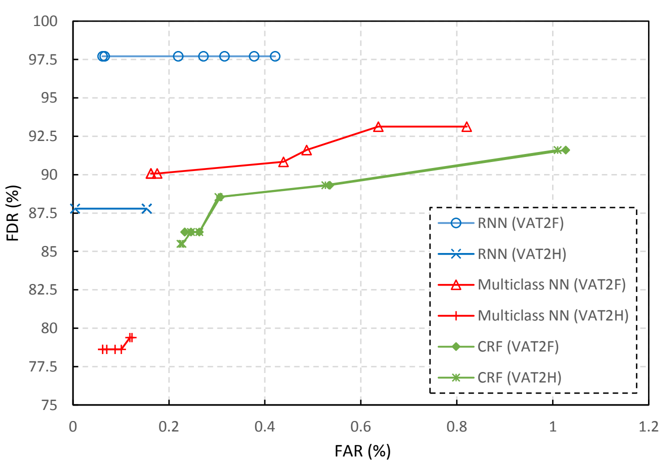


Fig. 6. Performance of CRF, RNN and Multiclass NN on the two voting-based failure detection algorithm VAT2H and VAT2F on dataset "W". The points on each line in the figure are obtained by the number of voters $N = 3, 7, 15, 21, 41, 45, 47$ from right to left.

Table 3 reports the overall FDR and FAR of the six models using the three datasets as test sets, which shows that our proposed method not only performs more steadily, but also has stronger universality. We see that RNN can obtain higher FDR and lower FAR simultaneously, compared with sequence-independent models.

RNN also outperforms other two sequence-dependent models: HMM and CRF. In general, HMM and CRF are effective on short-term dependent tasks, while the RNN is more suitable for long-term dependent tasks.

Another important variable is how long in advance we can detect an impending drive failure. The goal that hard drive manufacturers want SMART technology to achieve is more than 24 hours in advance. The average time in advance of our proposed RNN based failure prediction method is 241.6 hours (using VAT2F) and 208.6 hours (using VAT2H) on dataset "W", 494 hours (using VAT2F) and 494.9 hours (using VAT2H) on dataset "S", and 369.4 hours (using VAT2F) and 462.8 hours (using VAT2H) on dataset "M", which is sufficient for backing up data before the failure actually occurs.

4.6 Health Degree Assessment for Hard Drives

In this section, we evaluate the performance of the health degree assessments of RNN, CRF and Multiclass NN. First, we introduce the evaluation criteria for the health degree assessment results:

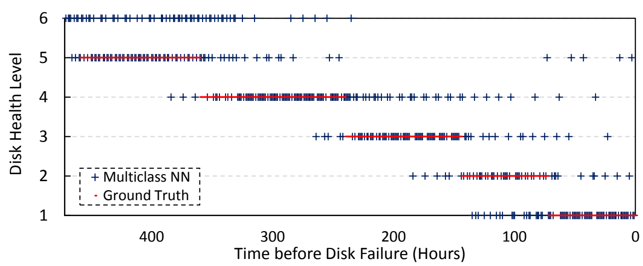
- H_{acc}: The accuracy of the health level assessment for all test disks at every time point.
- $H_{\text{acc}}^{\text{TSOL}}$ (tolerant skipping one level): H_{acc} with the added condition that we can tolerate assessment mistakes by one health level. For example, it is considered acceptable if a time point with health level 3 is assessed as health level 2 or 4.

$H_{\text{acc}}^{\text{TSOL}}$ gives a rough estimate of health degree so it is also valuable in practice. We perform case studies on health degree assessment problems. We randomly select 10 failed drives from test set of "W" and evaluated them using Multiclass NN, CRF and RNN. Figure 7 shows the health degree assessment results. From this figure, we find that RNN achieves higher assessment accuracy than the other two models. For the wrongly classified cases, the health degree assessment results of RNN are always clustered around the ground truth, while the wrongly assessed results of Multiclass NN and CRF are scattered.

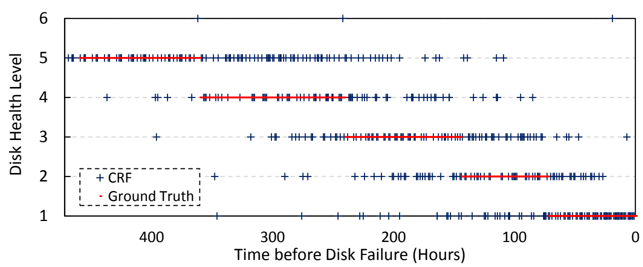
Table 4 reports the hard disk health degree assessment performance of Multiclass NN, CRF and RNN using the three datasets as test sets. We observe that the performance gaps between three models are very small on healthy drives. This is because it is not difficult for a classifier to identify a good drive, but it is hard to identify a failed drive and classify it into the right health degree before it breaks down. The overall hard disk health degree assessment performance above shows the effectiveness of RNN model, which clearly outperforms the sequence independent models and the short-term dependent models.

TABLE 4
Overall performance of Multiclass NN, CRF and RNN in terms of H_{acc} and $H_{\text{acc}}^{\text{TSOL}}$ on healthy drives and failed drives.

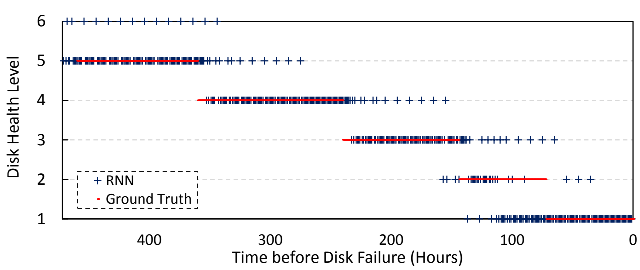
Methods	Drive Status	"W"		"S"		"M"	
		H_{acc} (%)	$H_{\text{acc}}^{\text{TSOL}}$ (%)	H_{acc} (%)	$H_{\text{acc}}^{\text{TSOL}}$ (%)	H_{acc} (%)	$H_{\text{acc}}^{\text{TSOL}}$ (%)
Multiclass NN	Healthy	99.19	99.40	99.84	99.94	99.40	99.73
	Failure	16.01	43.34	35.57	58.783	36.03	58.63
CRF	Healthy	99.57	99.59	99.97	99.98	99.98	99.98
	Failure	28.51	61.30	20.01	41.172	21.29	36.44
RNN	Healthy	99.73	99.93	99.91	99.99	99.66	99.97
	Failure	**41.05**	**64.86**	**37.30**	**60.80**	**61.75**	**90.934**



(a) Multiclass NN



(b) CRF



(c) RNN

Fig. 7. Health level assessment results of 10 randomly selected failed drives by using Multiclass NN, CRF and RNN.

5 Conclusion

In this paper, we propose a recurrent-neural-network-based model for predicting hard disk drive failure and giving health degrees, which treats the observed SMART attributes as time-sequence data. Experimental results show that our RNN based methods can achieve better performance than other sequence independent models and short-term sequence dependent models.

6 Acknowledgments

We would like to show our gratitude to Rebecca Stones for helping us improve the quality of language. We also thank Yuhan Jia for copying experimental data for us.

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] K. V. Vishwanath and N. Nagappan, "Characterizing cloud computing hardware reliability," in *IEEE Symposium on Cloud Computing*, June 2010.

[2] B. Schroeder and G. A. Gibson, "Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you?" *Conference on File and Storage Technologies*, 2007.

[3] Q. Xin, E. L. Miller, D. D. E. Long, S. A. Brandt, T. Schwarz, and W. Litwin, "Reliability mechanisms for very large storage systems," in *IEEE Conference on Mass Storage Systems and Technologies (MSST)*, 2003, pp. 146–156.

[4] J. F. Murray, G. F. Hughes, and D. Schuurmans, "Machine learning methods for predicting failures in hard drives: A multiple-instance application," *Journal of Machine Learning Research*, vol. 6, 2005.

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

[6] G. F. Hughes, J. F. Murray, K. Kreutz-delgado, S. Member, and C. Elkan, "Improved disk-drive failure warnings," *IEEE Transactions on Reliability*, vol. 51, 2002.

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

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

[9] B. Zhu, G. Wang, X. Liu, and D. Hu, "Proactive drive failure prediction for large scale storage systems," in *IEEE Symposium on Mass Storage Systems and Technologies (MSST)*, 2013, pp. 1–5.

[10] Y. Wang, Q. Miao, and M. Pecht, "Health monitoring of hard disk drive based on Mahalanobis distance," in *Prognostics and System Health Management Conference (PHM-Shenzhen)*, 2011.

[11] Y. Wang, Q. Miao, E. W. M. Ma, and K.-L. Tsui, "Online anomaly detection for hard disk drives based on Mahalanobis distance," *IEEE Transactions on Reliability*, vol. 62.

[12] T. Mikolov, S. Kombrink, L. Burget, J. Cernocky, and S. Khudanpur, "Extensions of recurrent neural network language model," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2011, pp. 5528–5531.

[13] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, and S. Khudanpur, "Recurrent neural network based language model," *In INTERSPEECH*, 2010.

[14] S. Kombrink, T. Mikolov, M. Karafit, and L. Burget, "Recurrent neural network based language modeling in meeting recognition," *INTERSPEECH*, 2011.

[15] M. Auli, M. Galley, C. Quirk, and G. Zweig, "Joint language and translation modeling with recurrent neural networks," in *Proc. Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2013.

[16] J. Li, X. Ji, Y. Jia, and B. Zhu, "Hard drive failure prediction using classification and regression trees," in *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2014, pp. 383–394.

[17] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[18] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 855–868, May 2009.

[19] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.