Machine Learning for Storage System Reliability Prediction

Yang Zhi Lin
Department of Computer Science and Engineering

the Chinese University of Hong Kong

Email: 1155092151@link.cuhk.edu.hk

***Abstract*—Disk failure is a critical issue for large scale storage systems in big data centres. In this project, I tried to implement different machine learning based methods presented in 3 papers in order to predict disk errors or failures. Finally, I compared these methods in a comprehensive view. It is expected that an integrated solution combining the merits of these methods can be proposed for service availability improvement in future.**

***Index Terms*—storage system reliability, machine learning**

I. INTRODUCTION

Reliability is crucial for storage systems in data centres, especially in the big data era. Frequent disk failure is a significant contributor to data centres' downtimes [1]. Existing methods for predicting impending drive failures include employing SMART (Self-Monitoring, Analysis and Reporting Technology) indicators to monitor internal disk attributes and reporting failures based on certain thresholds [2]. However, these thresholding algorithms have a very weak predictive power with only up to 50%~60% failure detection rates [3]. It has been shown that certain machine learning methods can demonstrate high performance for the prediction tasks [4][5][6]. In this project, I tried to reproduce three papers' results, which involve several machine learning methods as well as a variety of data processing techniques, to get an overall comparison of existing approaches.

II. LITERATURE REVIEW

Many studies have been conducted for predicting the reliability of storage systems. Botezatu *et al*. [4] came up with a machine learning-based analysis pipeline that can predict the necessity of a disk replacement. Their experiment included selecting informative SMART features, representing the time series data in a compact form, downsampling the imbalanced dataset and training classifiers to predict impending failures. In their experiment, the regularised greedy forests (RGF) showed the best overall performance. It was evaluated that their model can achieve up to 98% accuracy in predicting disk replacements.

Li *et al*. [5] explored the ability of Classification Trees (CT) to predict drive failures, and compared it with the state-of the-art model: Back propagation artificial neural networks (BP ANN) model. They evaluated their performances with different sets of features, different window sizes and different datasets. Experimental results showed that CT model could outperform BP ANN in prediction accuracy as well as stability and interpretability.

Xu *et al*. [6] developed a cost-sensitive ranking-based machine learning model for online prediction. They reviewed previous works and discussed two important issues: firstly, the extreme

imbalance of data generated by large-scale cloud service systems makes it hard to train a classification model; secondly, the cross validation method that is currently widely used is improper for practical use, since it means using part of the future data to train the model and testing the model with part of the past data, which is impossible in the real world online prediction scenario. To address those problems, they came up with a new method which utilised both system-level signals and disk-level SMART attributes. They also proposed a new feature selection method to select predictive features and a cost-sensitive ranking model to rank disks in terms of their error-proneness. They showed the effectiveness of their model and successfully deployed their approach in Microsoft Azure.

III. IMPLEMENTATION AND RESULTS

1. Paper name: 'Predicting Disk Replacement towards Reliable Data Centers'
(1) **Dataset:** At the first attempt, I used the open-source Blackblaze dataset for the whole year of 2017 from Q1~ Q4 and selected the model: ST4000DM000. There are about 35189 disks in total, among which there are 1061 failed disks.

(2) **Change point detection:** This paper [4] made an assumption that: 'when a SMART attribute is informative of disk replacement, we expect a significant shift in its values at some time point before the actual replacement'. However, this paper didn't clearly specify how they set the parameters of their change point detection model. In my experiment, I used the available online package of change point detection (source: https://medium.com/bigdatarepublic/contextual-changepoint-detection-with-python-and-r-using-rpy2-fa7d86259ba9).

(3) **Feature selection:** Following the paper [4], I then counted the SMART correlation frequencies for those features with change points: (correlation frequency: 'the percentage of drives for which a correlation with disk is observed') , the results are shown as follows:

Correlation frequencies

	percent
smart_1_normalized	30.62%
smart_1_raw	24.57%
smart_3_normalized	14.65%
smart_4_raw	33.08%
smart_5_normalized	6.11%
smart_5_raw	20.61%
smart_7_normalized	41.79%
smart_7_raw	61.91%
smart_9_normalized	54.25%
smart_9_raw	58.73%
smart_12_raw	33.08%
smart_183_normalized	12.75%
smart_183_raw	12.75%
smart_184_normalized	2.08%
smart_184_raw	2.08%
smart_187_normalized	33.93%
smart_187_raw	33.93%

smart_188_raw	0.89%
smart_189_normalized	2.36%
smart_189_raw	2.46%
smart_190_normalized	54.06%
smart_190_raw	57.99%
smart_192_raw	5.01%
smart_193_normalized	30.06%
smart_193_raw	58.33%
smart_194_normalized	53.99%
smart_194_raw	57.97%
smart_197_normalized	5.74%
smart_197_raw	44.71%
smart_198_normalized	5.74%
smart_198_raw	44.71%
smart_199_raw	0.38%
smart_240_raw	58.35%
smart_241_raw	48.02%
smart_242_raw	67.11%

In above table, I highlighted the features whose correlation frequencies are above 10% and selected them as the features to train the classifiers. There are 25 features in total.

(4) **Compact time series representation**: This paper [4] compacted every time series to a single value with the exponential smoothing method over a specific time window. In my experiment, I used the provided function: ewm in pandas library and the window sizes were chosen to be the median values of each attribute. After 4 stages of data preprocessing, I collected 15001 samples in total, with 458 replaced disk samples.

(5) **Downsample on healthy samples**: As the paper [4] suggested, the data is highly imbalanced between the two classes. So I used the K-means clustering algorithm in order to choose a representative subset of the data for the dense class: the healthy disks. Following the experiment setup, I ran K-means with 100 and 50 clusters as the parameters. For each cluster, I chose the top 10 data points closest to the centre of each cluster. After this step, there were totally 1000 healthy samples and 458 replaced samples to train ML classifiers.

(6) **Train Machine Learning Classifiers:** At this stage, I used 10-folds cross validation method to evaluate the performance of 6 types of machine learning classifiers and obtained the following results:

Healthy

	RGF	GBDT	RF	SVM	LR	DT
Precision	0.916 +/- 0.010	0.907 +/- 0.012	0.913 +/- 0.015	0.686 +/- 0.002	0.686 +/- 0.002	0.925 +/- 0.013
Recall	0.985 +/- 0.010	0.981 +/- 0.012	0.974 +/- 0.014	1.000 +/- 0.000	1.000 +/- 0.000	0.916 +/- 0.031
F-score	0.949 +/- 0.009	0.942 +/- 0.009	0.942 +/- 0.010	0.814 +/- 0.001	0.814 +/- 0.001	0.920 +/- 0.017

Replaced

	RGF	GBDT	RF	SVM	LR	DT
Precision	0.982 +/- 0.026	0.950 +/- 0.029	0.935 +/- 0.035	0.000 +/- 0.000	0.000 +/- 0.000	0.824 +/- 0.054
Recall	0.801 +/- 0.046	0.760 +/- 0.031	0.797 +/- 0.036	0.000 +/- 0.000	0.000 +/- 0.000	0.838 +/- 0.032
F-score	0.873 +/- 0.028	0.856 +/- 0.022	0.860 +/- 0.024	0.000 +/- 0.000	0.000 +/- 0.000	0.830 +/- 0.032

Clearly, RGF displayed the best overall performance on both the healthy class and the replaced class as described in the paper. However, LR classifier and SVM classifier showed zero precisions in my experiment. When I checked their prediction results, I found they actually predicted several replaced disks but the ratio was just too low, which led to an extremely low precision value. I think it may be because these two classifiers are more sensitive to class imbalance and may suffer from the overfitting. Then I changed the dataset to the 2015 whole year records and redid the whole experiment. This time I fixed some tiny errors in my code with reference to my partner Rui Zhe's code and then obtained the results as follows: This time the SVM still showed abnormal results, which suggested the instability of this type of algorithm.

stat

	GBDT	SVM	DT	LR	RF	RGF
P	0.912	0.334	0.875	0.723	0.905	0.934
R	0.911	1.000	0.891	0.688	0.887	0.906
F	0.911	0.500	0.883	0.699	0.896	0.919
Sd	0.060	0.001	0.045	0.097	0.058	0.050

2. Paper name: 'Hard Drive Failure Prediction Using Classification and Regression Trees'

(1) **Dataset**: The dataset I used can be downloaded from the following link: https://pan.baidu.com/share/link?shareid=189977&uk=4278294944 However, this is not the original dataset used in the paper [5], since the paper didn't present their data source. In this dataset, there are 23,395 drives in total, 433 failed drives and 22, 962 good drives. SMART attribute values were sampled from each working drive at every hour.

(2) **Data preprocessing**: The paper [5] dealt with the two classes of drives separately. Following the instructions, for each good drive, I randomly chose 3 samples from the earlier 70% of the samples within the week as the training data; the test data was 3 samples for each drive that were randomly chosen from the later 30% within the week. For each failed drive, I divided the data randomly into training and test sets by a 7 to 3 ratio, and took out the failed sample within a time window. After this, the training set contained 68886 good samples with 3636 failed samples; whereas the testing set contained 68886 good samples and 1560 failed samples.

(3) Feature selection: Following the paper[5], I tested the effectiveness of different numbers of features: 12, 19 and 13 features.

(4) Classification result: I also compared the performance between BP ANN and DT: using the following prediction rule: 'For each drive in the test set, look at all its samples in the test set

and predict failure as long as any of its sample is predicted as failed'[5]. The results are shown as follows:

My result:

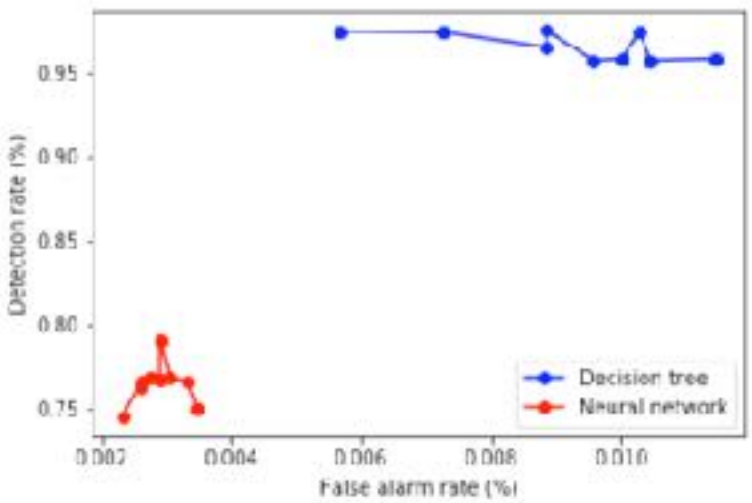
Model	Dataset	FAR (%)	FDR (%)
BP ANN	12 features	1.63	78.46
	19 features	2.52	83.72
	13 features	0.25	75.19
CT	12 features	0.46	91.54
	19 features	0.60	93.02
	13 features	0.60	93.02

Clearly, similar as the result in the paper[5], the classification tree showed the best overall performance with a low false alarm rate and a high failure detection rate. Meanwhile, in my experiment, when I tested BP ANN with the same set of parameters in different trials, the result varied largely, which suggested the instability of this type of ML algorithm.

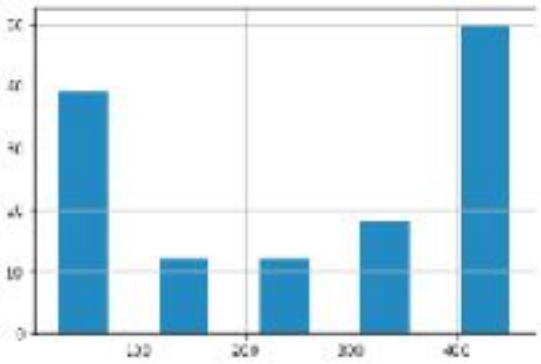
Then I varied the window size and obtained the following result:

Window Size	FAR (%)	FDR (%)
12 hours	0.35	92.25
24 hours	0.52	95.28
48 hours	0.52	92.86
96 hours	1.66	96.69
168 hours	1.48	94.55
240 hours	1.60	95.88

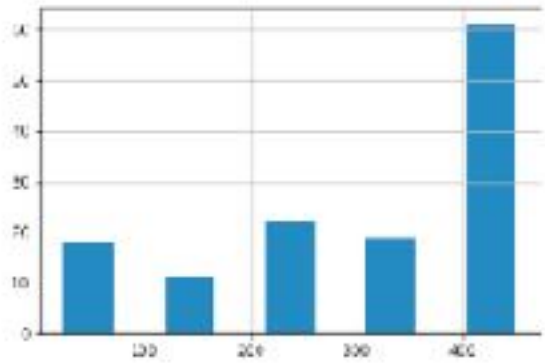
In addition, I used different number of voters and applied the following voting rule: 'When detecting a drive, we check the last N consecutive samples (voters) before a time point, and predict the drive is going to fail if more than N/2 samples are classified as failed, and the next time point is tested otherwise.' [5] And obtained the following result: (N=1,3,5,7,9,11,15,17,27)



Next, I calculated the Time In Advance (TIA), which is the time that the model is able to predict the failure in advance: (measured by hours)



Distribution of time in advance of BP ANN model.



Distribution of time in advance of CT model.

Lastly, I tested the robustness of 2 classifiers on datasets with different sizes:

Model	Dataset	FAR (%)	FDR (%)
BP ANN	A	0.377	57.627
	B	0.232	76.271
	C	0.015	66.949
	D	0.421	72.034
CT	A	0.842	84.034
	B	0.290	82.353
	C	0.682	89.076
	D	0.624	91.597

Combining above results, it is obvious that the classification tree algorithm is superior to BP ANN in terms of nearly every aspect of the performance.

3. Paper name: 'Improving Service Availability of Cloud Systems by Predicting Disk Error'

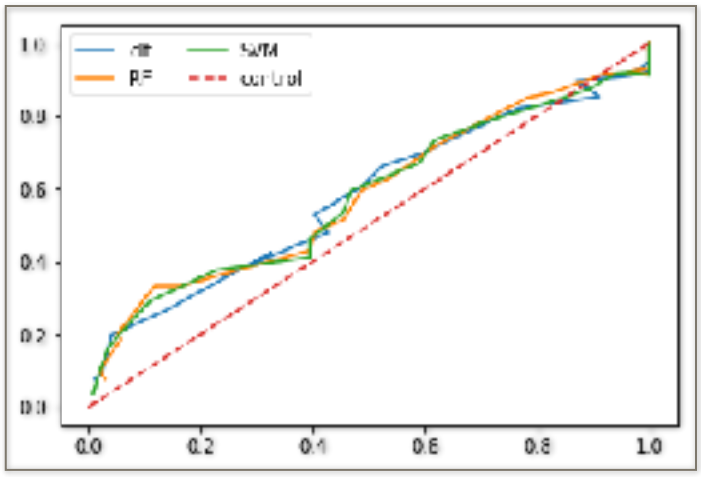
(1) **Dataset**: Since this paper [6] didn't provide their data source, I still used the Blackblaze dataset. I chose the data for the whole period of 4th quarter in 2017 and selected the model: ST4000DM000.

(2) **Labels**: Unlike the previous works which directly used the failure records as the labels to train models, this paper [6] utilised SMART 5: 'Reallocated Sectors Count' as the error indicator. They labeled data with 'the number of days between the data is collected and the first error is detected'. To use the prediction results of this regression task for the disk classification problem, they interpreted the predicted labels as the probability of being faulty and identify the failed disks accordingly.

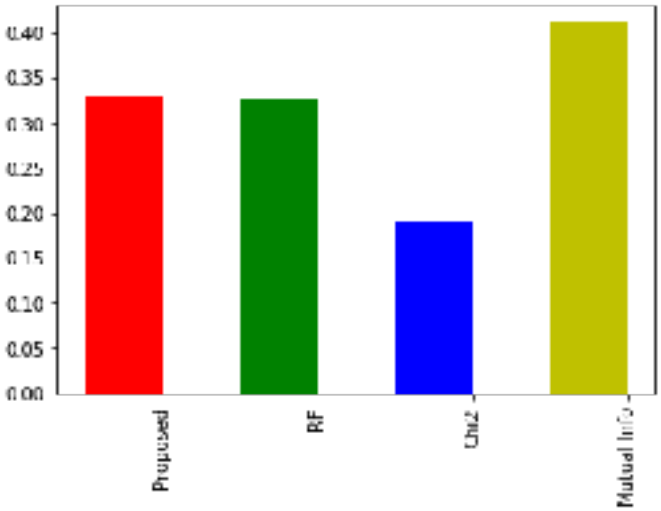
(3) **Feature identification**: this paper [6] collected two categories of features: SMART data and system-level signals. Due to the limitation of the data records in hand, I was only able to use the SMART attributes as features. Following the description of the paper, I created three types of statistical features as follows: the changes in a feature value over time—*Diff*, the variance of attribute values within a period—*Sigma* and the sum of attribute values within a window—*Bin.* The window sizes for these three features were chosen to be 3, 5 and 7. This paper identified 457 features in total from SMART and system-level data, while I created totally 96 features.

(4) **Feature selection**: this paper [6] proposed a feature selection method, which simulates the online prediction process, to prune away non-informative features. The basic process can be summarised as follows: divide the dataset by time into training set and test set, then iteratively delete every feature and evaluate the model on the validation set. If the prediction performance is improved, discard the feature. In my experiment, I simulated this process and removed 11 features.

(5) **Comparison of different classifiers**: This paper [6] adopted the FastTree algorithm which is a form of "Multiple Additive Regression Trees" (MART) gradient boosting algorithm. Due to the inaccessibility of the FastTree package offered by Microsoft, I used another member in MART family: the Gradient Boosting Regressor, which is available in skicit-learn library. Following the paper, I varied the threshold of prediction, which in turn varied the resulting ROC curve, and compared it with Support Vector Machine (SVM) and Random Forest (RF) methods. The plotting is shown below: in my experiment, the performances of these three methods are quite close.



(6) **Comparison of different feature selection methods**: Following the paper [6], I compared three conventional feature selection methods: Chi-Square, Mutual Information, and Random Forest with the proposed CDEF method. I plotted the True Positive Rate (TPR) corresponding to the False Positive Rate (FPR) that is in the range of 0.1 to 0.2. The picture is shown below:



IV. DISCUSSION

In this project, I tried to reproduce the results in 3 papers concerning different machine learning methods to predict disks failures, as well as several feature engineering methods, and compared their performances. Finally, I draw the conclusions from my experiment results as follows:

(1) Generally, tree-based machine learning methods are far superior to other types of machine learning algorithms in predicting the disk failure, especially those ensemble method such as Gradient Boosting Decision Tree and Gradient Boosting Regressor, concerning both the scores of metrics and the prediction robustness.

(2) Apart from the choice of machine learning model, the feature engineering is also a crucial factor that will vastly affect the final results. But which type of feature engineering method is the most suitable really depends on the specific situation.

(3) The quality of dataset is also an influential factor that determines the final result. This is part of the reason I failed to achieve the exactly equal result as the paper suggested. Thus, it is worthwhile to study the transfer learning method in the future, which enables the use of existing trained prediction model on new disks model.

V. REFERENCE

[1] Data center downtime costs. http://www.emerson.com/en- us/News/Pages/ Net-Power-Study-Data-Center.aspx.

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

[3] B. Zhu, G. Wang, X. Liu, D. Hu, S. Lin, and J. Ma, "Proactive drive failure prediction for large scale storage systems," *2013 IEEE 29th Symposium on Mass Storage Systems and Technologies (MSST)*, 2013.

[4] M. M. Botezatu, I. Giurgiu, J. Bogojeska, and D. Wiesmann, "Predicting Disk Replacement towards Reliable Data Centers," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 16*, 2016.

[5] J. Li, X.P. Ji, Y.H. Jia, B.P. Zhu, G. Wang, Z.W. Li, X.G. Liu, 'Hard Drive Failure Prediction Using Classification and Regression Trees,' *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, Atlanta, GA*, 2014, pp. 383-394.

[6] Y. Xu, K.X. Sui, R. Yao, H.Y. Zhang, Q.W. Lin, Y.N. Dang, P. Li, K.C. Jiang, W.C. Zhang, J.G. Lou, M. Chintalapati, D.M. Zhang, 'Improving Service Availability of Cloud Systems by Predicting Disk Error,' in *Proceedings of the 2018 USENIX Annual Technical Conference*, 2018.