Machine Learning for Storage System Reliability Prediction

Author: Ruizhe LI | **Supervisor:** Patrick P. C. Lee
Department of Computer Science and Engineering
The Chinese University of Hong Kong

ABSTRACT

Nowadays, disk failure is one of the most excruciating troubles that IT departments have been suffering from. Despite several traditional methods like RAID have been invented to defend disk failures, availability and reliability of the system are still hugely affected. Machine learning, as one of the most popular prediction techniques, has also been used in the prediction of such failure, aiming at accuracy improvement and cost reduction. Although several machine learning prediction techniques have been proposed, different methods use different models and procedures, thus have different performance.

In this paper, we rebuild the methods described in three different papers. These papers use machine learning methods to enhance disk failure prediction accuracy in different ways. The basic idea is using the current disk state, e.g. the current SMART attributes or system signals, to predict disk failures in advance, so that the nearly broken disks could be detected, then further actions are able to be applied to reduce company loss. Based on the SMART attributes given by Backblaze, the result of each paper is tested, and the strength and limitation of each method are analyzed.

1. INTRODUCTION

In the past few years, disk errors have been becoming one of the most intractable and costly problem. Disk errors could cause large scale service failure, user complaint, and huge revenue loss. Based on a study conducted on 63 data center organizations in the U.S, the average downtime cost from 2010 to 2016 has increased by 38%, from $505,502 to $740,357 [1]. Studies based on Facebook and Google data show that 20-57% of solid state drives experience at least one sector error [2, 3]. In these days, machine learning has becoming more and more popular in many industrial areas. To cope with disk failure problem, many machine learning methods have been proposed [4, 5, 6, 7, 8]. The current methods are mainly focusing on the prediction of disk failure with cross-validation. These methods share similar goals, but they are implemented in different ways.

In this paper, three different methods are validated: Predicting Disk Replacement towards Reliable Data Centers (ADF [9]), Improving Storage System Reliability with Proactive Error Prediction (ATC17 [10]), Improving Service Availability of Cloud Systems by Predicting Disk Error (ATC18 [11]). The feature selection method, the efficient down sampling method, and the high accuracy disk failure prediction model proposed by ADF [9] are tested in this paper. ATC17 [10] focuses on the prediction of the probability of the occurrence of sector error in the near future. Its model is tested with analysis about the most important feature in prediction. In last part of the paper, part of ATC18 [11] is validated according to

our understanding. ATC18 [11] focuses on the application of the online machine learning concept. It builds a regression model to rank the error-proneness of each disk. The feature selection method described in ATC18 [11] is also tested. In the implementation of above papers, the strengths, limitations, and challenges are elaborated. We expect in the future, these methods could be delicately combined and expanded to produce an even better result.

2. PAPER IMPLEMENTATION

In this section, we present the re-implementation procedure of each paper. All of the following sub-sections are based on the SMART attributes in Backblaze dataset. And they are divided by papers. There are four fundamental sections for each paper: Goal – the paper's primary goal; Method – the summary of prediction method described in each paper and the implementation details based on our understanding of the paper; Results and Comparison – the results from our own experiment following their methods, and the comparison with the results described in the paper; Discussions – the challenges we have encountered during the re-implementation, and the possible reasons and explanations of the differences between our results and the original results in the paper. Because the procedure of each step has been elaborated in detail in each paper, here we only provide an outline and the implementation according to our understanding.

2.1. Predicting Disk Replacement towards Reliable Data Centers (ADF [9])

- Label: Failure value recorded in the Backblaze dataset
- Features: Selected and Compacted SMART attributes

2.1.1. Goal

- Provide informative SMART attributes for disk replacement
- Apply machine learning method on the selected attributes to predict impending replacements with high accuracy (81-98%)

2.1.2. Methods

2.1.2.1. Selection of relevant SMART attributes

This section aims at automatically selecting the attributes most informative to the disk failure. The paper assumes that, before a disk failure happens, some attributes may show a permanent, significant change. And these changepoints forebode a coming failure. The attributes with such changepoint should be selected as the features as informative indicators for later training. Thus, there are two steps in this section: changepoint detection and permanency validation.

a) Changepoint Detection

Since the probability distributions for each attribute are quite different and sophisticated, we are not able to specifically ascertain the exact distribution for each feature. All the SMART attributes are assumed to be normally distributed. Changepoint detection is to find the shift points that divide the time series into two different distribution functions, e.g. there might be a significant shift in mean value after the changepoint indicating a different normal distribution. In order to find out the correlation between disk failure and the changepoint detection, this process is

only applied to failed disks.

b) Permanency Validation (omitted)

After the statistical test for changepoint, we have to verify the permanency of the change to eliminate the accidental, restorable changepoints. Thus, the part of timeseries after the changepoints should be tested against the value obtained from the healthy disks. However, because the method described in the paper is rather ambiguous, and the changepoint elimination is not so contributive to the prediction accuracy, this process is omitted here.

c) Attributes Selection

After the permanent changepoints have been sifted out, on each attribute we compute the percentage of the number of failed samples with a changepoint with the total number of failed samples. And only the attributes with a relevancy larger or equal to 1% are selected.

2.1.2.2. Compact time series representation

Since the sequence information of the timeseries is quite useful to predict disk failures in advance, we compact timeseries to a point representation through following steps:

a) Find the median of the distribution of the time stamps of their corresponding changepoint.

b) Use the median as the width of window to compact the selected features. The EWMA function in the pandas package is applied on the selected features, with the median value being used as the span.

2.1.2.3. Class balancing via informative down sampling

The dataset is highly imbalanced. Only few disks are failed in the dataset. Take the ST4000DM000 disk model (Sgt A) as an example. There are only 824 failed disks among 29908 in total. If training is applied equally on the whole data, the prediction model will be prone to the healthy disks to achieve a high accuracy. To solve the imbalance problem, down sampling is needed. We use the K-Means clustering algorithm provided by the sklearn package to reduce the number of healthy disks. By obtaining the samples around the cluster centers of the K-Means algorithm, the most typical healthy samples could be sifted out. After the down sampling process, we control the ratio between failed and healthy disks to be 1 : 2.

2.1.2.4. Classification for disk replacements

As stated in ADF [9], GBDT (Gradient Boosting Decision Tree), SVM (Support Vector Machine), DT (Decision Tree), LR (Logistic Regression), RF (Random Forest), RGF (Regularized Greedy Forest) have been used in prediction. All the abovementioned machine learning algorithms except for the RGF are called directly from the sklearn package, and the RGF algorithm is installed as an API. The evaluation metrics are precision score, recall score, and f1 score implemented in the sklearn metrics.

2.1.3. Results

Dataset: Backblaze 2017 Q1 – 2018 Q1, 15 months in total

Model: ST4000DM000 (Sgt A)

Serial number	Percentage	Serial number	Percentage	Serial number	Percentage	Serial number	Percentage
S242_raw	57.1%	S193_norm	28.0%	S198_raw	44.3%	S189_raw	2.1%
S7_raw	55.4%	S1_raw	22.5%	S197_raw	44.3%	S189_norm	2.0%
S193_raw	52.9%	S5_raw	20.5%	S241_raw	41.3%	S184_norm	1.9%
S9_raw	50.6%	S3_norm	13.5%	S7_norm	38.4%	S184_raw	1.9%
S240_raw	50.2%	S183_raw	12.8%	S187_norm	33.7%	S188_raw	0.89%
S190_norm	48.7%	S183_norm	12.8%	S187_raw	33.7%	S199_raw	0.49%
S190_raw	48.6%	S198_norm	11.2%	S4_raw	30.8%	S4_norm	0.08%
S194_raw	48.6%	S197_norm	11.2%	S12_raw	30.7%	S192_norm	0
S194_norm	48.6%	S5_norm	6.5%	S1_norm	29.1%	S241_norm	0
S9_norm	46.8%	S192_raw	4.5%	S3_raw	0	S10_raw	0
S199_norm	0	S242_norm	0	S191_norm	0	S191_raw	0
S10_norm	0	S12_norm	0	S240_norm	0	S188_norm	0

Table 1: Sgt A feature correlation

a) Feature-Failure Correlation:

SMART Number	Feature Name	Value in ADF [9]	Value in experiment
3	Spin-Up Time	NA	13.5%
4	Start/Stop Count	Not in paper	30.8%
9	Power-On Hours	Not in paper	50.6%
12	Power Cycle Count	Not in paper	30.7%
183	SATA Downshift error Count or Runtime Bad Block	0.5%	12.8%
192	Power-off Retract Count, Emergency Retract Cycle Count	Not in paper	4.5%

Table 2: Features selected in our experiment but not in original paper

Explanation: Figures in Table 1 is higher than figures in ADF [9]. The omission of changepoint permanency validation could be the main reason. Some temporary, accidental changepoints are also counted in the correlation. In short, the noise of attributes pushes the figures up.

Table 2 shows the SMART attributes that are only selected in our experiment based on the 1% correlation threshold, but not included in ADF [9]. SMART 4, 9, 12, 192 are not mentioned in ADF. Perhaps the researchers thought these features are less relevant to the failure prediction, because they are time-related-only, which means their value is constantly changing as time flowing. Excluding these features has no significant influence on prediction.

b) Prediction Accuracy

	RGF	GBDT	RF	SVM	LR	DT
F1	0.988	0.990	0.986	0.990	0.897	0.991
Recall	0.986	0.986	0.992	1.0	0.870	0.989
Precision	0.990	0.993	0.980	0.980	0.948	0.993

Table 3: Precision, Recall, F-score of different classifiers in our experiment

		RGF		GBDT		RF		SVM		LR		DT	
		SgtA	HitA	SgtA	HitA	SgtA	HitA	SgtA	HitA	SgtA	HitA	SgtA	HitA
Replaced	P	0.98	0.84	0.97	0.82	0.93	0.82	0.93	0.72	0.73	0.72	0.89	0.74
	R	0.98	0.79	0.96	0.78	0.94	0.76	0.95	0.65	0.81	0.59	0.87	0.61
	F	**0.98**	**0.81**	**0.96**	**0.80**	**0.94**	**0.79**	**0.94**	**0.68**	**0.77**	**0.65**	**0.88**	**0.67**
	Sd	**0.01**	**0.02**	0.01	0.04	0.05	0.08	0.02	0.05	0.07	0.1	0.04	0.03
Healthy	P	0.99	0.93	0.98	0.92	0.97	0.92	0.97	0.87	0.89	0.85	0.94	0.86
	R	0.98	0.95	0.98	0.94	0.96	0.93	0.96	0.90	0.85	0.90	0.95	0.91
	F	**0.98**	**0.94**	0.98	0.93	0.97	0.92	0.96	0.88	0.87	0.87	0.94	0.88
	Sd	**0.01**	**0.02**	0.02	0.03	0.04	0.05	0.02	0.04	0.08	0.05	0.02	0.02

Table 4: Precision, Recall, F-score of different classifiers in the original paper

This method is also tried on different disk models. Results are shown in the following table:

Dataset: Backblaze 2015 Q1 – 2015 Q4, 12 months in total

Model: ST4000DM000, ST31500541AS, Hitachi HDS722020ALA330, Hitachi HDS5C3030ALA630

	# total	# failure	# failure percentage
ST4000DM000	29908	824	0.03
ST31500541AS	1970	271	0.14
Hitachi HDS722020ALA330	4737	159	0.03
Hitachi HDS5C3030ALA630	4634	74	0.02

Table 5: Overview statistics of other disk models

model	Metrics	GBDT	SVM	DT	LR	RF	RGF
ST4000DM000	P	0.99	0.98	0.92	0.60	0.99	0.99
	R	0.95	0.96	0.87	0.61	0.95	0.94
	F	0.97	0.97	0.89	0.60	0.97	0.97
	Sd	0.02	0.01	0.03	0.30	0.02	0.02
ST31500541AS	P	1.00	1.00	0.90	0.34	1.00	1.00
	R	0.92	0.92	0.76	0.97	0.92	0.92
	F	0.96	0.96	0.82	0.50	0.96	0.96
	Sd	0.01	0.01	0.07	0.02	0.01	0.01
Hitachi HDS722020ALA330	P	0.99	0.93	0.89	0.71	0.99	0.99
	R	0.86	0.87	0.86	0.85	0.85	0.85
	F	0.92	0.89	0.86	0.77	0.91	0.91
	Sd	0.04	0.07	0.09	0.08	0.04	0.04
Hitachi HDS5C3030ALA630	P	1.00	0.97	0.63	0.46	1.00	1.00
	R	0.71	0.71	0.74	0.56	0.71	0.69
	F	0.83	0.82	0.67	0.50	0.83	0.81
	Sd	0.06	0.07	0.05	0.05	0.06	0.04

Table 6: Precision, Recall, F-score of different classifiers when the method is applied on different disk models

Explanation: As shown in the Table 6, this method can also achieve a high accuracy with low standard deviations in other disk models. RGF, SVM and GBDT are the most efficient machine learning methods. They can achieve 80% accuracy in all disk models listed above. Because models have different number of failure disks as shown in Table 5, the prediction accuracies are not same among different models. There is a clear trend that, the models with more failed disks can achieve a higher prediction accuracy.

2.1.4. Discussions

2.1.4.1. Similarities

a) Order of accuracy – Tree models are better predictors

Take Sgt A as an example. The orders of accuracy of different machine learning model are similar: RGF >= GBDT >= RF >= SVM >= DT >= LR. These consistency means RGF, GBDT, RF models do have some strengths on answering this kind of questions. The algorithms behind "tree structure" can properly interpret relationship between health status and the selected SMART attributes of each disk.

b) Disk models' influence – More failed disks lead to better performance

And the result also indicates some differences among the accuracies of several disk models. If we focus on the RGF model, order of accuracies of each model is Sgt A > Sgt B > Hit A > Hit B, which is consistent with the descending order of the number of failed disks, and this order seems not related to the total number of disks and failed percentage. Down sampling could be one of key reasons for this relationship. In the experiment, the data is sampled to be commensurate to failed number. Therefore, the size of training set is not determined by the whole data set, but the number of failed disks.

2.1.4.2. Differences

a) RGF's performance – Performance is not as overwhelming as in the original paper

In ADF [9], RGF is a dominant machine learning method. It shows a conspicuous advantage over other methods. On the contrary, this huge gap did not show up in my result. GBDT, SVM, RF, and RGF can all reach a F1 score around 97%, in Sgt A model. This could be interpreted with different attributes we selected.

b) Six more selected attributes – Attributes are more comprehensive

I selected 6 more attributes than the paper. And based on my understanding, why these attributes were not shown in the paper is because either some attributes were not accessible at that time, or the author thought they were irrelevant. And more attributes may lead to a higher prediction accuracy.

c) Higher feature-failure correlation

The feature-failure correlation percentage is higher in the experiment than in ADF [9]. This perhaps is because the permanency validation is omitted in our experiment, thus temporary, unstable, accidental changepoints may be counted. In that case, a higher correlation is shown. An obvious example is the SMART 9 Power-On Hours is counted as an informative attribute. As in the dataset, the Power-On Hours of a disk should be constantly increasing. There should be no sudden change in such time-related attribute.

d) Dataset discrepancy – A larger dataset is used in ADF [9].

In addition to attributes, another difference between the author and I is our datasets was not completely same. A larger dataset is used in the paper, from 2013 to 2015 excluding several months in between, totally 27 months' data. However, my dataset only contained the data from 2015 whole year, 12 months' data. Thus, the accuracy dropped in my result could be explained by the decrement of dataset size.

2.2. Improving Storage Reliability with Proactive Error Prediction (ATC17 [10])

- Label: 1 if SMART_5_raw (S5) increases in the next week, else 0.
- Features: SMART attributes and their increasement.

2.2.1. Goal

To predict whether a drive will have a sector error within a given time interval, based on its past behavior.

2.2.2. Methods

2.2.2.1. Observation on overview statistics of disk models

Since the procedure of sector error measurement is not clearly shown to readers, in our experiment all the disks with a positive value on S5 will be considered as suffered from sector error. Then, we counting the number of the disks suffered from such error and divided by the number of all disks to get percentage. The result is shown in Table 7 in Results section.

2.2.2.2. Data preprocessing

The ATC17 [10] paper does not use the provided disk failure number as labels,

instead, it generates its own label. As the prediction target is whether there will be an error occurs in the following week, the label is set to the increasement of S5 in a week, i.e. the difference of S5 value between next Sunday and this Sunday. Then the label is set to be 1 if the difference is positive, otherwise the label is 0. It is possible that the S5 value's difference is negative. In this case, the value is also set to 0. As it is a minor case, it does not affect our experiment significantly

The training input is the SMART attributes given in the ATC17 [10] paper and their increasement, just before the testing week, i.e. use the SMART values on this Sunday and their increasement during this week to predict whether there will be a sector error occurring in next week.

At last, data is normalized using min-max-scaler given in the sklearn package.

2.2.2.3. Down sampling

In the ATC17 [10] paper, the random down sampling is used to solve the imbalanced problem. In our experiment, the K-Means clustering method is applied to get a better down sampling performance. The detailed procedure is same as the ADF [9] down sampling section. Note that, the number of healthy disks is reduced to 3 times the number of the failed disks, instead of 2 times in ADF [9].

2.2.2.4. Classifiers training

As stated in the ATC17 [10] paper, SVM (Support Vector Machine), CART (Classification And Regression Tree), LR (Logistic Regression), RF (Random Forest), NN (Neuron Network) have been used in prediction. All the above-mentioned machine learning algorithms are called directly from the sklearn package. The evaluation metrics are precision score, recall score, and f1 score implemented in the sklearn metrics.

2.2.3. Results

Dataset: Backblaze 2015 Q1 – 2015 Q4, 12 months in total
Model: ST3000DM000
Input features: S1, S4, S5, S7, S9, S12, S187, S193, S194, S197, S199, S4_increase, S5_increase, S7_increase, S9_increase, S12_increase, S187_increase, S193_increase, S197_increase, S199_increase
Note that the ATC 17 does not specify the time period for their data. 2015 is the latest year with ST3000DM000 model, which is used in the original paper.

Overview

	Models	smart_5_raw	smart_187_raw	smart_196_raw	smart_197_raw	Capacity (TB)	# Drives
0	ST4000DM000	0.007381	0.015470	0.000000	0.015066	4.0	29670.0
1	ST3000DM001	0.120719	0.177226	0.000000	0.070205	3.0	1168.0
2	Hitachi HDS5C3030ALA630	0.032349	0.000000	0.032132	0.012375	3.0	4606.0
3	Hitachi HDS722020ALA330	0.133675	0.000000	0.133675	0.036515	2.0	4683.0
4	Hitachi HDS5C4040ALE630	0.015038	0.000000	0.015414	0.007519	4.0	2660.0
5	HGST HMS5C4040ALE640	0.006032	0.000000	0.006032	0.002665	4.0	7129.0
6	HGST HMS5C4040BLE640	0.000967	0.000000	0.000967	0.002578	4.0	3103.0

Table 7: Overview statistics

Explanation: Take ST3000DM001 (Sgt) disk model as an example. According to our experiment, there are 12.07% of total disks have a positive S5 value. And the Hitachi HDS722020ALA330 (Hit) has 13.37% disks suffering from sector errors. This figure is close to 11% described in the ATC17 [10] paper, which indicates sector error is one of the most common problem that different disk models and manufacturers are facing.

Classification result:

	CART	SVM	NN	LR	RF
P	0.968137	0.991304	0.992000	0.900943	0.984615
R	0.865217	0.906522	0.930797	0.939493	0.974638
F	0.911131	0.946601	0.943269	0.915293	0.979045
Sd	0.089723	0.018153	0.052120	0.046580	0.012678

Table 8: Different machine learning methods applied on ST3000DM001

Explanation: Around 500 samples are used in training and validation. RF achieves the best performance with an accuracy of 97.9% under 0.0127 standard deviation. This result is consistent with the ATC 17 paper, where RF is the best machine learning model as well.

Feature importance:

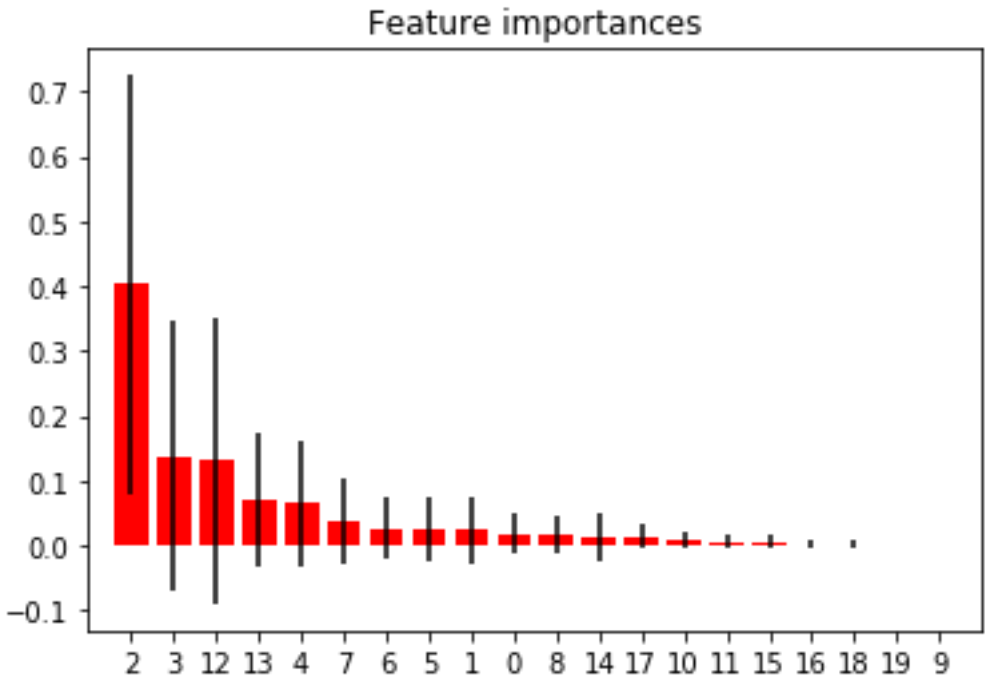

Figure 1: Feature importance under RF model

Explanation: There are 20 features in total with the following order: S1, S4, S5, S7, S9, S12, S187, S193, S194, S197, S199, S4_increase, S5_increase, S7_increase, S9_increase, S12_increase, S187_increase, S193_increase, S197_increase,

S199_increase, ranging from 0 to 19. As shown in Figure 1, S5 has the most significant influence on the prediction. S7 (Seek Error Rate) is the second important feature. They are followed by their increasement. Other features have little influence on the decision. This means that S5 and S7 are the dominative factors. A disk with S5 or S7 errors is more likely to encounter such errors again in the next week.

Even on different disk models, the S5 and S7 are the most important features for sector error prediction:

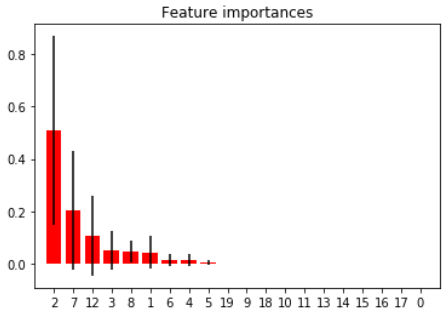

Figure 2: ST4000DM000

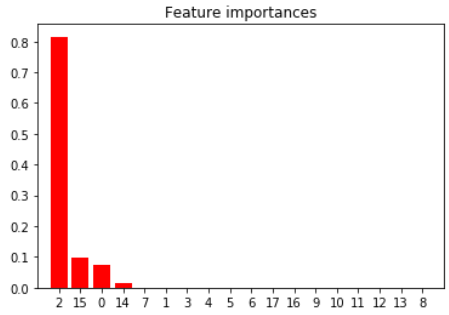

Figure 3: Hitachi HDS722020ALA330

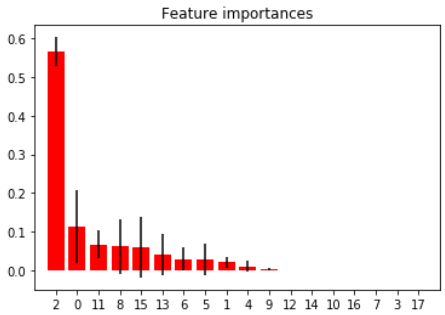

Figure 4: Hitachi HDS5C3030ALA630

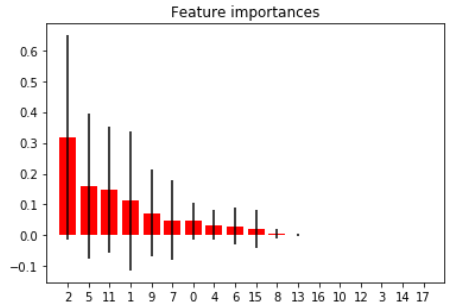

Figure 5: Hitachi HDS5C4040ALE630

2.2.4. Discussions

2.2.4.1. Similarities

a) High accuracy with small dataset

Different sizes of dataset are used in our experiment. Certainly, a higher accuracy could be achieved when the dataset is large enough. However, the size of dataset is not a determinative factor. A dataset as small as 500 samples can also achieve a high accuracy above 90%. The prediction accuracy is not highly related to the size of dataset.

b) RF is the best prediction method

As shown in Table 8, the RF consistently outperform or match the performance of other classifiers. And it is the easiest and fastest machine learning method with few parameters to tune. The number of trees is not sensitive to the accuracy of prediction. This is consistent with ATC 17.

2.2.4.2. Differences

a) Low feature correlation assumption

In ATC17 [10] paper, the author assumes that the input features are nearly unrelated to each other. However, in our experiment, we showed that there is a

huge correlation among features. Take ST3000DM000 as an example. The original number of features is 20 in total. After applying PCA on original inputs, only 9 features remind. Fewer features push the efficiency up, while at the same time maintaining a high performance.

b) NN, SVM, and LR's bad performance
As shown in Table 8, though the performances of these algorithms are not as good as RF, they can achieve a 90% accuracy as well.

2.2.4.3. Challenges

The implementation procedure described in ATC17 [10] is rather ambiguous. The author only gives out the general idea. Many details are not clear to readers. Following are unclear points:

a) Dataset
In ATC17 [10], the authors only indicate that the experiments use SMART attributes as input, but how large the dataset is and when the data is collected are unknown to readers. Thus, we choose to use the latest data containing the specified models, which is Backblaze data in 2015.

b) Label
What is going to be predicted is another primary problem. As written in the ATC17 [10] passage, they are predicting whether there will be an error within the next 7 days. What should the "error" be? How to generate label according to this definition is unclear to readers.

c) Input features
As stated in ATC17 [10], the explanatory variables should be all SMART parameters reported by a drive as possible candidates for explanatory variables. After the preprocessing, only 11 key features are kept. However, the feature selection process is ambiguous to readers. Why this set of features are selected as input?

2.3. Improving Service Availability of Cloud Systems by Predicting Disk Error (ATC18 [11])

- Label: The number of days between the data is collected and the first error is detected
- Features: Selected SMART attributes, and Diff, Sigma, and Bin of each selected attributes.

2.3.1. Goal

- Provide a feature engineering method for selecting stable and predictive features.
- Construct a ranking model to increase the accuracy of cost-sensitive online prediction.

2.3.2. Methods (CDEF – Cloud Disk Error Forecasting)

2.3.2.1. Feature engineering

a) Label Preparation

ATC18 [11] does not use the binary failure label given in the Backblaze dataset. Instead, they obtain the disk label through root cause analysis of service issues by field engineers. This procedure is hidden from us.

In order to predict error-proneness, a regression method is preferred. The given label definition is: The number of days between the data is collected and the first error is detected. Since the "first error" is not clarified, we assume the sector error (S5) is the error it refers to. Thus, the label is calculated as the difference between the date when the disk has with a positive S5 for the first time, and the sample's own date.

b) Feature Identification
In ATC18 [11], the selected SMART attributes are informative to coming disk errors. Both SMART records and system signals, e.g. File System Error, are used in the paper. However, in our experiment, only SMART attributes are used, so that we could test the efficiency of the method when features are limited.

In addition to directly collected raw data, three statistical features are also used:

Diff: Difference of a feature over a period (3 days in experiment)

Bin: Sum of the attribute valuers over a period (5 days in experiment)

Sigma: Variance of attribute values over a period (7 days in experiment)

These features contain time information and change trend of each attribute. As usually disk failure is slowly generated rather than a sudden event, the information containing in time series is critical to failure proneness prediction.

Note that, this step could be easily accomplished using pandas python package.

c) Feature Selection
As there are three kinds of statistical features and one set of raw features, the input dimension is rather high. Many features are redundant or irrelevant to the prediction. Thus, feature selection is necessary.

In the ATC18 [11], a pretty simple method is applied in feature selection: a loop. First, the whole dataset is divided into two parts TR1 and TR2 strictly by time (this is important because time information may influence prediction result and leads to a high but impractical performance). Then, for each feature f in selected attributes, we train models on TR1 without feature f, and test the model on TR2. If testing result is higher when feature f is removed, we permanently remove this feature. The pruning process keeps going until all features are tested. The remaining features are the selected ones.

This method is rather simple and straight. It can prune out unrelated features and leave the informative ones. However, it is inefficient, because for each loop the machine learning model has to be trained again, and the parameters are also needed to be tuned to get the best performance under the current set of features. This process costs quite a lot of time, especially when a huge dataset is used.

2.3.2.2. Cost-sensitive ranking model

In this section, we train a prediction model to rank the error-proneness of disks.

a) Machine Learning Model:
FastTree algorithm, which is a form of MART (Multiple Additive Regression Trees), is applied as the regression method in ATC18 [11].

As Microsoft ML document says, FastTree is an efficient implementation of the MART gradient boosting algorithm. Thus, the Gradient Boosting Regressor in sklearn python package is applied in our implementation.

The ranking process produces the predicting days before the first error occurs. The disks with bigger error-proneness are in the front of sequence. The top r results returned by the ranking model are considered as the faulty ones. And r is determined by the optimal value according to the following metric.

Note that, in order to implement the online machine learning technique, cross-validation is not applied in training. And the testing data is always late than the training data.

b) Metric:
$Cost = Cost1 * FP_r + Cost2 * FN_r$
Where:
FP_r: False Positive rate = FP / N = FP / (FP + TN)
FN_r: False Negative rate = FN / N = FN / (FN + TP)
Cost1 is the cost of wrongly identifying a healthy disk as faulty. Cost2 is the cost of failing to identify a faulty disk. The values of Cost1 and Cost2 are empirically determined by experts in product teams. Since our dataset is totally different from the one used in ATC18 [11] which belongs to Microsoft, the Cost1 and Cost2 values are meaningless in our case. Theoretically, this cost-sensitive method is efficient in reducing the loss of disk failure. However, as the datasets are different, and cost is unspecified, this method is not implemented in our experiment.

2.3.3. Results

Dataset: Backblaze 2017 Q3, 3 months in total
Model: ST4000DM000
Feature Selection:

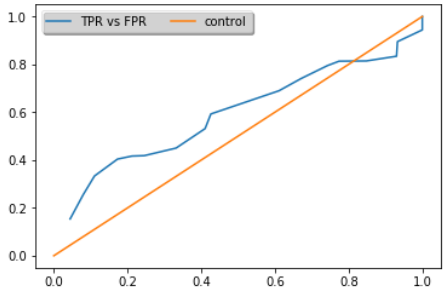

Figure 6: Proposed feature selection

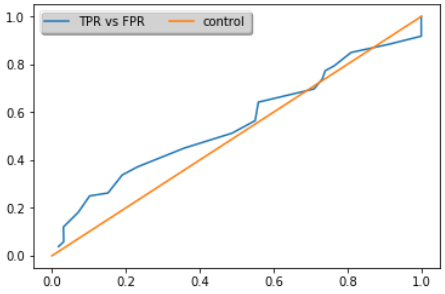

Figure 7: RF feature selection

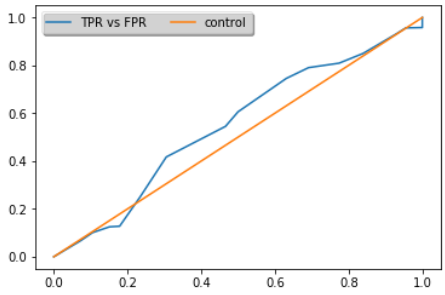

Figure 8: Chi2 feature selection

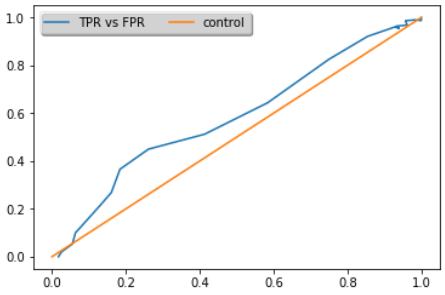

Figure 9: Mutual info selection

Note: For Figure 6-9, x-axis is FP_r, y-axis is FN_r

Explanation: The above Figures are the results of prediction using different feature selection methods.

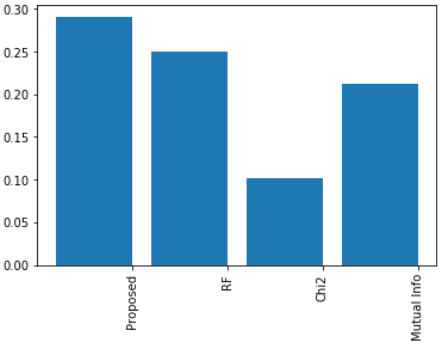

Figure 10: Feature selection comparison

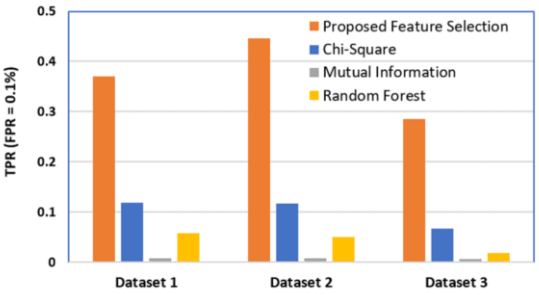

Figure 11: Feature selection in paper

Explanation: As what is shown in Figure 10 and Figure 11, we get a similar result that the "Proposed Feature Selection" achieved a better score than the other three. However, it does not show an advantage as large as shown in ATC18 [11]. And in ATC 18, the performance of feature selection methods follows the order of Proposed Feature Selection, Chi-Square, Random Forest, and last Mutual Information. However, the order in our implementation is: Proposed Feature Selection, Random Forest, Mutual Information, and last Chi-Square. The orders are not same.

ML Model Comparison:

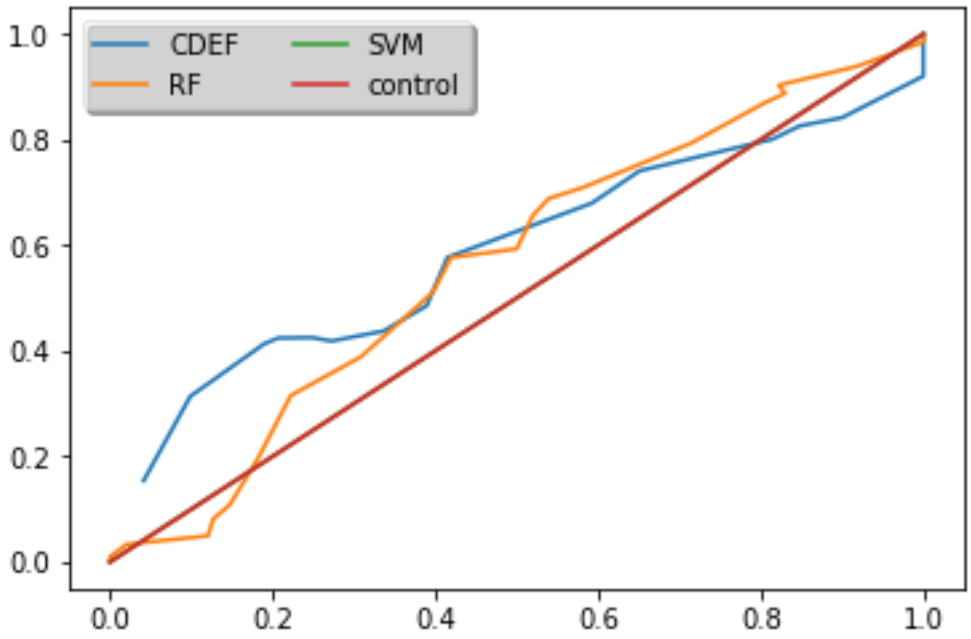

Figure 12: Performance of Different model test

Explanation: As shown in Figure 12, the CDEF (FastTree or MART) method have no clear advantage over other machine learning methods.

2.3.4. Discussions

2.3.4.1. Challenges

ATC18 [11] has many ambiguous points in its implementation. Thus, we meet many challenges and assumptions during implementation:

a) Training and Validation Data Ratio

Problem: This paper only used one month's data. It may be sufficient if the data is attracted from the extremely huge Microsoft database. However, the only data accessible to us is the BlackBlaze dataset. And it is obviously insufficient for machine learning. For another thing, the paper does not illustrate a reasonable ratio for training and validation. How much data we should use to validate our model is an unanswered question.

Solution: Up to now, we still do not have a sound solution for this problem. In my implementation, I divided the dataset to 2:1, 2 months for training and 1 month for validation.

b) "First Error" Definition

Problem: The term "First Error" is rather ambiguous. It can indicate any disk error, such as read error (smart 1), reallocate sector count (smart 5), SATA downshift error count (smart 7), so on and so forth, or just simply a disk failure. Indeed, there is a plausible solution to integrate all of these errors and as long as one of it goes up we say an error occurs. However, more than half smart attributes indicate some kind of error. If we use all of them to build labels, then few left to be used as input.

Solution: As this paper is a descendent of ATC 17, I used SMART 5 as the single label indicator. If SMART 5 is larger than 0, then we say an error has occurred on the disk.

2.3.4.2. Similarities

a) Proposed feature selection has the best performance
According to Figure 10, the proposed feature selection, which is based on the "loop reduction", has the best performance. On the contrary, other selection methods, which are mainly based on the correlation among features, do not perform well. However, since a in the proposed feature selection, every feature has to be tested once, this method is quite straightforward but rather slow. It is not practical with large size of data.

b) Online prediction accuracy is much lower than the cross-validation result
The accuracy of CDEF method proposed in ATC18 [11] is much lower than the results given in other two papers. This is mainly because of the online machine learning. In ATC18 [11] the model is trained on the earlier part of data and later tested on the later part. Thus, time information is excluded in this implementation. Although the accuracy of online prediction is pretty low, it is much closer to the reality, where we have no idea about future.

2.3.4.3. Differences

a) The order of the feature selection efficiency
Although the proposed feature selection has the best performance in our experiment, it does not show an overwhelming advantage. Moreover, the order of other feature selection methods is not consistent with ATC18 [11].

b) CDEF model does not show any advantage over other methods
As Figure 12 shows, the FastTree algorithm does not show any advantage over other machine learning methods. Here are some possible reasons:

- Procedure: As abovementioned, this paper has several points unclear to readers. Thus, the implementing procedure may not be exactly same as the paper.
- Dataset: The data used in the paper contains both smart attributes and system signals. And the sizes of datasets are different either.
- ML method: Although fasttree is an efficient implementation of MART algorithm, it is still possible that there are some optimal steps included in the Microsoft ML Server. Since I used the sklearn GradientBoostingRegression, it might not be implemented in the same way as Microsoft FastTree.

3. CONCLUSION

Disk failure is one of the most intractable troubles that IT departments face. In this paper, we implement three disk failure prediction methods. Each of them has strengths and limitations.

ADF [9] uses the SMART attributes to build model to predict disk error in

advantage, so that the disks could be replaced before failure actually happens. The changepoint detection technique for feature selection is quite reasonable and practical. As SMART attributes for different disk models have different distributions, this method could be expanded with more accurate statistical probability model. Another interesting point of ADF [9] is the K-Means down sampling method, which solves the imbalanced dataset problem by representing the dominative class with most typical samples. The method described in ADF [9] could reach very high accuracy. However, it suffers from the bias from cross-validation. It makes use of the time information in the future in prediction.

ATC17 [10] attempts to prediction the sector error occurrence in near future (a week). Our results show that this method could reach a very high accuracy in different disk models. Moreover, the most important feature in prediction is S5 (Sector Error Count), which means one sector error has occurred indicates more sector errors are coming. And the results for ATC17 [10] also suffers from the cross-validation bias.

ATC18 [11] proposed a CDEF method to rank the disk error-proneness. This paper is unique in the following ways:

1. No down sampling is involved.
   Down sampling technique is applied in ADF [9] and ATC17 [10]. However, in ATC18 [11], it ranks the disk error-proneness and uses a number of disks with the highest error-proneness as the failed disks. The number of disks to be regarded as failed is determined by a cost-sensitive function. However, as this function is constructed by experts in Microsoft, we have no way to test its efficiency.

2. Self-designed feature selection
   ATC18 [11] does not use any popular feature selection technique. Instead, it tries to remove each feature to see whether there is an improvement on results. Although the model trained with the feature selected by this method has a higher performance, the “looping” method is inefficient especially when dataset is large.

3. It uses a regression model to rank the disk error-proneness.
   In other two papers, classification method is used to tell disks with errors from healthy disks in advance. However, ATC18 [11] uses a regression model. Thus, it has very low accuracy.

4. Cross-validation is not applied.
   In order to get rid of the bias of cross-validation, ATC18 [11] applied online machine learning method. However, the details of online machine learning method have not been elaborated. And the final accuracy is rather low.

Many points in ATC18 [11] are unclear to readers. Its method heavily depends on experts and dataset. Thus, the validity of CDEF method proposed in the paper is hard to be proved.

4. Reference

[1] PONEMONINSTITUTE. Cost of data center outages, 2016. https://planetaklimata.com.ua/instr/Liebert_Hiross/Cost_of_Data_Center_Outages 2016 Eng.pdf.

[2] MEZA, J., WU, Q., KUMAR, S., AND MUTLU, O. A large-scale study of flash memory failures in the field. In Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (New York, NY, USA, 2015), SIGMETRICS '15, ACM, pp. 177–190.

[3] SCHROEDER, B., LAGISETTY, R., AND MERCHANT, A. Flash reliability in production: The expected and the unexpected. In Proceedings of the 14th Usenix Conference on File and Storage Technologies (Berkeley, CA, USA, 2016), FAST'16, USENIX Association, pp. 67–80.

[4] GOLDSZMIDT, M. Finding soon-to-fail disks in a haystack. In Proceedings of the 4th USENIX Conference on Hot Topics in Storage and File Systems (Berkeley, CA, USA, 2012), HotStorage'12, USENIX Association, pp. 8–8.

[5] PINHEIRO, E., WEBER, W.-D., AND BARROSO, L. A. Failure trends in a large disk drive population. In Proceedings of the 5th USENIXConferenceonFileandStorageTechnologies(Berkeley, CA, USA, 2007), FAST '07, USENIX Association, pp. 2–2.

[6] PITAKRAT, T., VAN HOORN, A., AND GRUNSKE, L. Acomparison of machine learning algorithms for proactive hard disk drive failure detection. In Proceedings of the 4th International ACM Sigsoft Symposium on Architecting Critical Systems (New York, NY, USA, 2013), ISARCS '13, ACM, pp. 1–10.

[7] WANG, Y., MIAO, Q., MA, E. W. M., TSUI, K. L., AND PECHT, M. G. Online anomaly detection for hard disk drives based on mahalanobis distance. IEEE Transactions on Reliability 62, 1 (March 2013), 136–145.

[8] ZHU, B., WANG, G., LIU, X., HU, D., LIN, S., AND MA, J. Proactive drive failure prediction for large scale storage systems. In 2013 IEEE 29th Symposium on Mass Storage Systems and Technologies (MSST) (May 2013), pp. 1–5.

[9] 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.

[10] 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.

[11] 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.