Weekly Report 6

Paper: **Hard Drive Failure Prediction Using Classification and Regression Trees (DSN)**
Supplementary paper: **Proactive Drive Failure Prediction for Large Scale Storage Systems**

1. dataset overview

(1) Source: collected from a single running datacenter of Baidu Inc.

(2) Statistics: 23,395 drives in total, 433 failed drives and 22, 962 good drives

(3) SMART attribute: 12 attribute values sampled from each working drive **at every hour**

2. data preprocessing

(1) For each good drive, training data: randomly choose **3 samples** from the **earlier 70%** of the samples within the week; test data: randomly choose **3 samples** from the **later 30%** as test data within the week

(2) For each failed drive, divide them randomly into training and test sets in a **7 to 3 ratio,** and take out the failed sample within a ***time window***, that is, **the last *n* hours before the failure actually occurs**

(3) Dataset description:

Training set: 68886 good samples, 3636 failed samples

Testing set: 68886 good samples, 1560 failed samples

3. Feature selection

3 strategies:

1)12 features: 10 normalised values with 2 raw values

ID #	Attribute Name
1	Raw Read Error Rate
2	Spin Up Time
3	Reallocated Sectors Count
4	Seek Error Rate

5	Power On Hours
6	Reported Uncorrectable Errors
7	High Fly Writes
8	Temperature Celsius
9	Hardware ECC Recovered
10	Current Pending Sector Count
11	Reallocated Sectors Count (raw value)
12	Current Pending Sector Count (raw value)

2) 19 features (used by the supplementary paper):

Add 7 change rates of 6 hours: #1, #5, #187, #195, #197 and RAW attributes #5 and #197

3) 13 features (used by the target paper):

Exclude the 10th feature "Current Pending Sector Count" and the 12th feature the raw value of "Current Pending Sector Count", add 3 change rates of 6 hours: "Raw Read Error Rate", "Hardware ECC Recovered" and "Reallocated Sectors Count (raw value)".

4. ML classification Result:

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**

FAR: False alarm rate=false positive/(false positive+true negative)

FDR: Failure detection rate=true positive/(true positive+false negative)

The result in paper:

Model	Dataset	FAR (%)	FDR (%)
BP ANN	12 features	0.44	89.47
	19 features	0.25	90.23
	13 features	0.20	90.98
CT	12 features	0.57	95.49
	19 features	0.63	94.74
	13 features	0.56	95.49

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**

Comparison:

1) The classification tree demonstrates excellent performance which is very close to the result in the paper.

2) BP ANN gets a fairly comparable but lower result.

3) My observation: BP ANN is not a stable algorithm, its result can differ between different training turns even if all hyper parameter remains true

5. Compare between different time window:

Result in paper:

Time Window	FAR (%)	FDR (%)
12 hours	0.31	93.98
24 hours	0.33	93.98
48 hours	0.39	95.49
96 hours	0.21	96.24
168 hours	0.09	95.49
240 hours	0.11	93.23

My 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**