Extracting information for failure prediction from intermittent data

Balakrishna S Kesavan[1] and Amol B Mahamuni[2]
[1] Principal Data Scientist, IBM India
[2] Program Director, IBM India

e-mail: balakrishna.kesavan@ibm.com

Abstract. Edge devices record data intermittently and device failures are rare. These make it hard to extract patterns of device failures using common temporal and multivariate approaches, like boosted trees. Survival analysis can produce actionable insights in such scenarios.

Key words. Survival analysis, failure prediction, predictive maintenance

1. Introduction:

With Industry 4.0 gaining acceptance, there will be an increase in the number of intelligent devices, like Industrial IoT (IIoT) devices, that will be used at the edge locations of businesses. Many use cases for edge analytics have been identified at factories, retail stores, warehouses, hotels, distribution centers, and in vehicles. In all of these, the edge device itself will need predictive maintenance, so that the intended use case is reliably supported.
Edge devices record data intermittently for reasons like the devices being powered down during non-working hours, they may have limited storage space or their network connections may be unreliable. Device failures are also rare. This makes device failure prediction modeling challenging.

2. The challenge – intermittent data and very rare failures:

To explain the problem and proposed solution better, we use S.M.A.R.T. (Self-Monitoring, Analysis and Reporting Technology) data about Hard Disk Drive (HDD) failures, hosted for public use by Black Blaze[1]. The SMART dataset has 3+Mn observations, covering 69 HDD models, of which only 215 observations show HDD device failures. The observations were recorded once a day starting from 1st Jan 2016 to 29th Apr 2016. There are 94 features (sensor readings), in addition to the 'failure' flag. The data contributors, Black Blaze, have found the following features, in Table 1, to be useful for failure prediction[2]. Plus, we learn that as the value of SMART_5_Reallocated Sectors Count goes up, the probability of device failure goes up[3]. These inputs are important points of collaboration with device engineering.

SMART 5		Reallocated Sectors Count
SMART 187		Reported Uncorrectable Errors
SMART 188		Command Timeout
SMART 197		Current Pending Sector Count
SMART 198		Uncorrectable Sector Count

Table 1. Features with engineering relevance to failure prediction

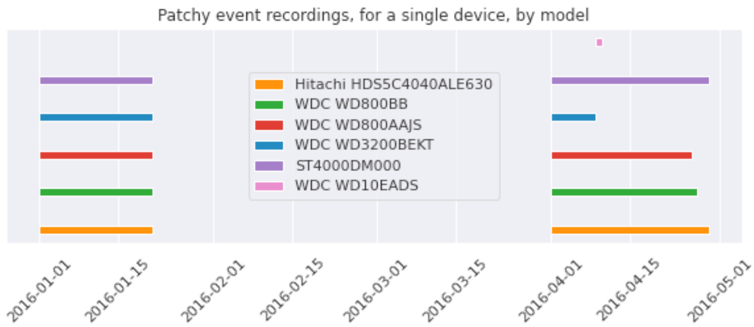

Fig 1. Intermittent observations

To illustrate the intermittent data problem, we plot details for a few devices in Fig 1. As seen, data records are not continuous and don't cover the same periods. All further analysis is limited to the same 6 models.

model	# of observations	failure	failPercent
WDC WD3200BEKT	2	1	50.000000
WDC WD800AAJS	28	6	21.428571
WDC WD800BB	6	1	16.666667
WDC WD10EADS	256	2	0.781250
ST4000DM000	63946	131	0.204860
Hitachi HDS5C4040ALE630	5271	3	0.056915

Fig 2. Dataset is imbalanced both by number of observations by model and failures

Fig 2. shows the imbalanced nature of the dataset. Typical imbalanced dataset treatment techniques involve some combination of over-sampling the minority class and under-sampling the majority class. But this approach is not helpful with this dataset. For example, one could use the Synthetic Minority Oversampling Technique (SMOTE) for creating additional records in the minority class. But this technique relies on having a minimum number of minority class records that form a "feature space", within which new records are synthesized. By default, this is set to six neighbors. But as seen in the Fig 2, there are instances where the number of observations and failures are less than six. Some of them are so few that there is a serious risk of injecting fake patterns if we were to synthesize data.
Similarly, under-sampling the majority class using techniques like Tomek Links is difficult when the number of records of devices that haven't failed are too few.

3. Feature engineering:

As part of feature engineering, the temporal features ‘Remaining Useful Life’ (RUL) and ‘Failure within time window’ (FIW = 1 week) were created as the dependent feature for regression and classification respectively, as seen in Fig 3. The intent is to discover temporal patterns of degradation across all the features (sensor readings).

serial_number		date	failure	RUL	FIW	smart_5_normalized
9VY8TC9H	2765	2016-04-08	0	10	0	61
	2766	2016-04-09	0	9	0	58
	2767	2016-04-10	0	8	0	55
	2768	2016-04-11	0	7	1	53
	2769	2016-04-12	0	6	1	53
	2770	2016-04-13	0	5	1	50
	2771	2016-04-14	0	4	1	48
	2772	2016-04-15	0	3	1	44
	2773	2016-04-16	0	2	1	40
	2774	2016-04-17	0	1	1	36
	2775	2016-04-18	1	0	1	28

Fig 3. Feature engineering to derive dependent features RUL and FIW

In Fig 4, the correlation between the dependent and the independent features is explored but none were found. Some of the independent features have high correlation with each other.

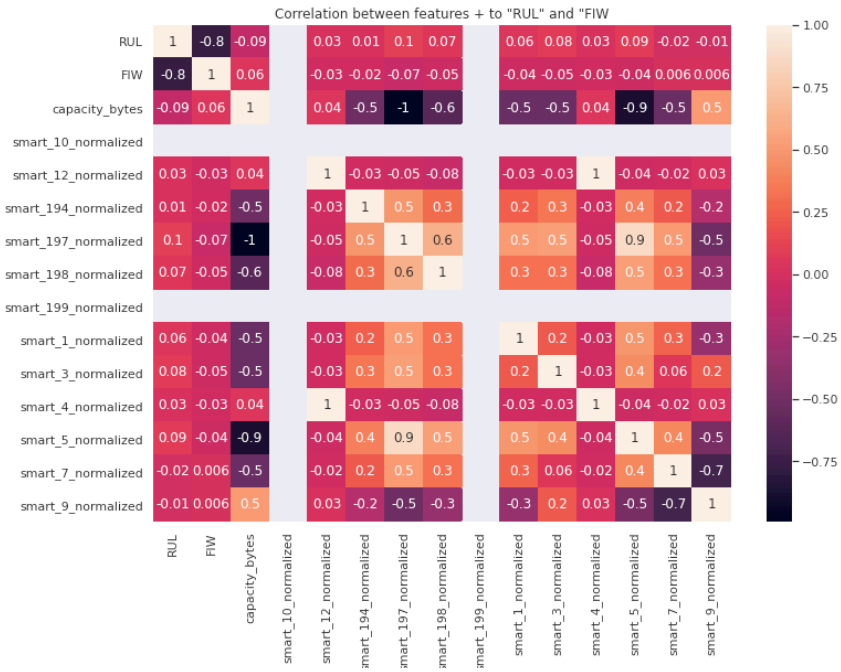

Fig 4. Correlation heatmap between suggested features to the dependent features, RUL and FIW

4. Multivariate analysis:

Boosted tree models for regression (to predict RUL) and classification (to predict FIW) were built, after adding device model names as dummy features. As shown later, the failures rates vary widely between models, making this an important feature. Also, we ignore the correlation between independent features since trees in general are not much affected by such correlation[4].

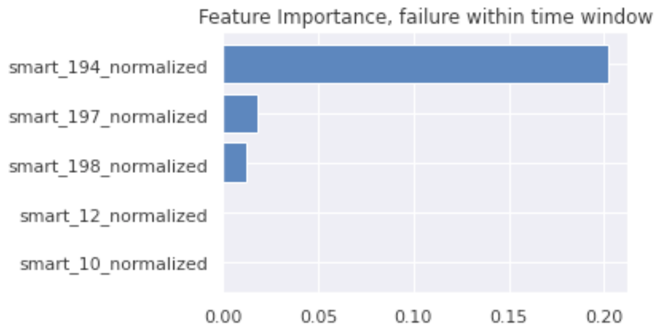

Fig 5. Near zero feature importance in boosted tree classification model

The boosted tree regression model showed a poor fit with neg_mean_squared_error = -1025.642, standard deviation = 43.681 for RUL, i.e. the mean error is about 32 days when the median value of RUL is 21 days. The classification model also showed poor fit with cross validated f1 score mean = 0.242, standard deviation = 0.047. (The target f1 score is 1.0). The feature importance plot in Fig 5, from the boosted tree classification model, shows that even the top 5 features have very poor contribution, explaining the low f1 score. Only 2 of the features that Black Blaze suggested, for hard disk engineering reasons, appear here (smart 197 and smart 198). Smart 194 has 77 distinct values compared to 6 & 5 for smart 197 and smart 198 respectively. This is the likely reason for the inclusion of this feature, which was already flagged as non-informative by Black Blaze.
In conclusion, multivariate boosted trees that include a temporal dependent feature showed poor fit.

5. Squeezing out insights using survival analysis:

Where the above temporal and multivariate approach struggles, by using the univariate Kaplan Meier algorithm for survival analysis, we can still obtain actionable insights out of this dataset.

serial_number	start	end	model	failure	duration
WD-WX71A9290300	2016-01-01	2016-01-21	WDC WD3200BEKT	0	20
WD-WX71A9290300	2016-04-01	2016-04-09	WDC WD3200BEKT	1	8

Table 2. Partially processed observations for the model “WDC WD3200BEKT”

Consider the partially processed observations for the model “WDC WD3200BEKT” in Table 2. There are only 2 of them in the dataset, of which only the last observation shows a failure.

```
These are the failures for model WDC WD3200BEKT:
           removed  observed  censored  entrance  at_risk
event_at
0.0              0         0         0         2        2
8.0              1         1         0         0        2
20.0             1         0         1         0        1
```

Table 3. Survival table

Using observations like that in Table 2, a survival table, in Table 3, extracts information about how many devices of each model were being tracked ("at risk") at different points in time ("event_at", in days), how many of these recorded no failures during or at the end of the study ("censored") and how many failed ("observed"). Survival analysis uses information even from devices that never failed and hence, were "censored". Instead of discarding observations relating to devices that didn't fail, censoring allows using all available information. That is, the fact that both "WDC WD3200BEKT" devices survived day 0 and 1 device has survived ***past*** day 20 is used in survival probability computation.

The survival probability[5, 6] is calculated using this formula where 'n' is the number of devices at risk ('at_risk') just before time 't'('event_at') and 'd' is the number of deaths ('observed') at 't'.

$$\hat{S} = \prod_{t_i < t} \frac{n_i - d_i}{n_i}$$

Day	Number of devices at risk	Number of devices that failed	Survival probability (product of terms)
0	2	0	(2-0)/2 = 1
8	2	1	{(2-0)/2}x{(2-1)/2} = 0.5
20	1	0	{(2-0)/2}x{(2-1)/2}x{(1-0)/1} = 0.5

Table 4: Survival probability calculations (Refer Table 3 for the variable values)

Insight 1 – planning predictive maintenance schedules: The survival function plot, in Fig 6, provides an easy to interpret view of survival probabilities. For example, for model "WDC WD3200BEKT" the median probability of survival is 50% by the 8th day. Using this information, predictive maintenance schedules can be set. For example, inspect/ maintain/ replace components before median probability of survival drops below 50%.

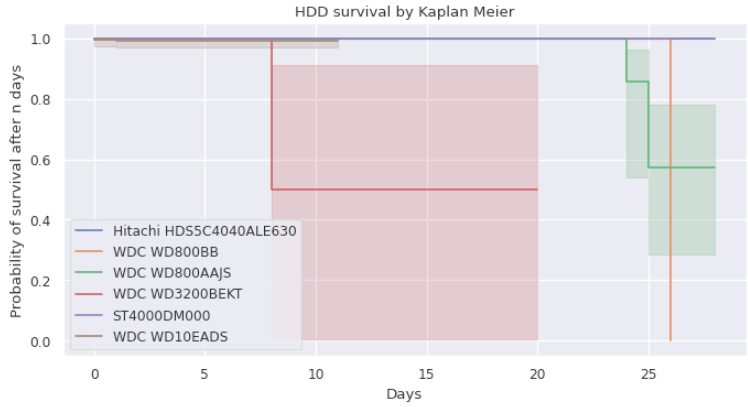

Fig 6. Survival function plot showing survival probability over time

The plot also shows the 95% confidence interval of the estimates. If the desire is to maintain even the lower confidence interval above 50%, then, model "WDC WD800AAJS" would need to be maintained by the 25th day.

Insight 2 – statistical significance of the differences between models. A simple pair wise log rank test, in Fig 7, can tell if the survival curves of individual models are meaningfully different.

t_0	-1
null_distribution	chi squared
degrees_of_freedom	1
test_name	logrank_test

		test_statistic	p	-log2(p)
Hitachi HDS5C4040ALE630	**ST4000DM000**	5.41	0.02	5.64
	WDC WD10EADS	18.72	<0.005	16.01
	WDC WD3200BEKT	751.37	<0.005	547.10
	WDC WD800AAJS	799.60	<0.005	581.94
	WDC WD800BB	262.02	<0.005	193.36
ST4000DM000	**WDC WD10EADS**	24.96	<0.005	20.70
	WDC WD3200BEKT	389.80	<0.005	285.82
	WDC WD800AAJS	621.79	<0.005	453.49
	WDC WD800BB	90.00	<0.005	68.51
WDC WD10EADS	**WDC WD3200BEKT**	13.30	<0.005	11.88
	WDC WD800AAJS	0.22	0.64	0.65
	WDC WD800BB	0.05	0.83	0.27
WDC WD3200BEKT	**WDC WD800AAJS**	14.00	<0.005	12.42
	WDC WD800BB	3.00	0.08	3.59
WDC WD800AAJS	**WDC WD800BB**	0.47	0.49	1.02

Fig 7. Pair wise log rank test

The logrank test statistic is calculated from the differences between the observed deaths for a group and expected deaths, under the null hypothesis that all groups share the same survival curve, summed across all ordered death times.[7,8]

For instance, there is statistically significant difference between the models referred to in Insight 1 ('WDC WD3200BEKT' and 'WDC WD800AAJS').

Insight 3 – comparing device model survival at a future point in time: The Restricted Mean Survival Times (RMST), shown in Fig 8, another measure from survival analysis, is the area under the survival curve up to a prespecified time horizon. All else being equal, the model with the larger RMST has the higher probability of surviving at a given point in time (day 50 in this case).

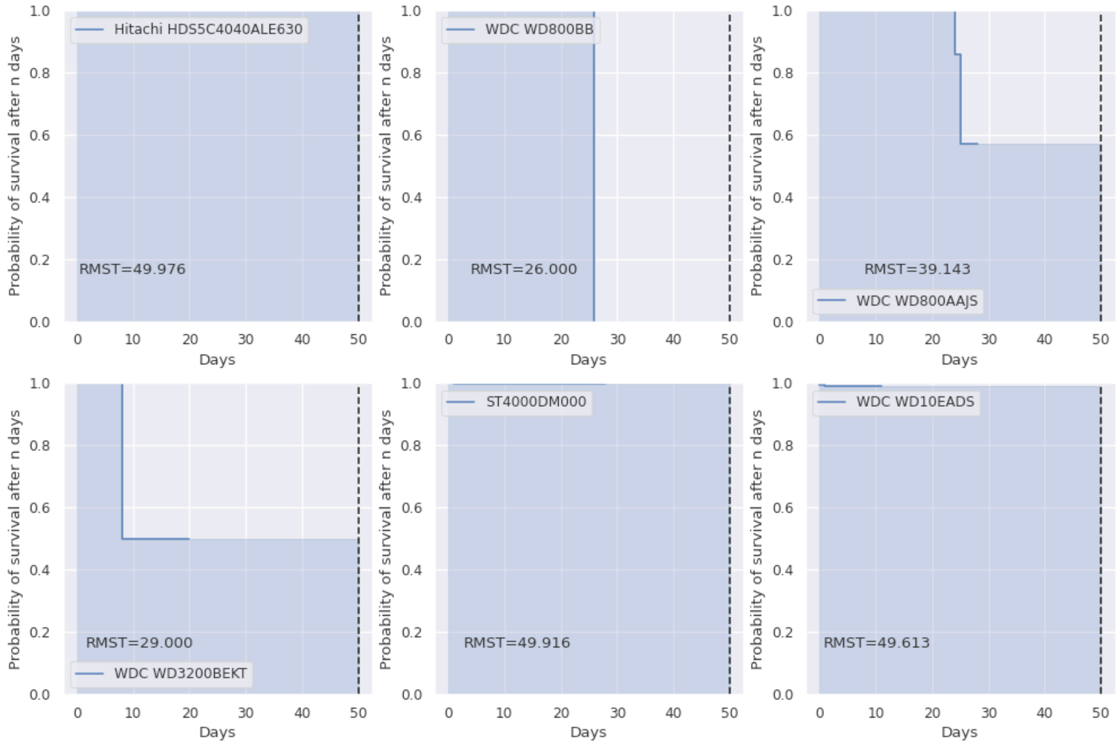

Fig 8. Comparing survival of models at a future point in time

Survival regression using Cox Proportional Hazard could only be fitted after excluding the feature ‘model’, representing the HDD model. But this is an important feature, as seen in other analysis, and its’ exclusion is inappropriate. Hence, this is not reported.

6. Conclusion:

Typical edge device logs have intermittent data, very rare failures and the features may not be informative. Temporal, multivariate approaches, like boosted trees are not helpful in such cases. However, survival analysis is able to provide actionable insights.

7. References:

[1] Data source: https://www.kaggle.com/backblaze/hard-drive-test-data

[2] Black Blaze feature recommendation: https://www.backblaze.com/blog/what-smart-stats-indicate-hard-drive-failures/

[3] Source: https://en.wikipedia.org/wiki/S.M.A.R.T.

[4] Source: https://datascience.stackexchange.com/questions/31402/multicollinearity-in-decision-tree

[5] “Survival Analysis A Self-Learning Text” by David G. Kleinbaum and Mitchel Klein, published by Springer.

[6] Python Lifelines: https://lifelines.readthedocs.io/en/lat est/index.html

[7] Source: https://lifelines.readthedocs.io/en/latest/Examples.html#transforming-observational-data-into-survival-table-format

[8] Log rank test explained: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC403858/