Abstract: Data centers generally use hard drives as data storage device. Large companies heavily rely on data and use many hard drives, which become challenging to monitor manually. When there is an issue with the hard-disk, it should function for at least next 24 hours for the data back-up to be done. But in ideal cases, the hard-disk fails even before 24 hours resulting in loss of data. Hard drive failures cause data loss which can cause a serious problem for the users. As backup, multiple copies of data can be stored in the system, but it might increase the cost at the same time. In industries, hard drives are monitored by setting threshold for several critical metrics.

SMART (Self-Monitoring, Analysis, and Reporting Technology) attributes of hard disks can be useful in detecting the failure rate of the hard-disks. It is useful to predict the hard drive failure by developing a model, so that it can be used to get useful insights to improve the system reliability and help cut cost.

Keywords: Hard Drive Failure Prediction; Data Center Predictive Maintenance; Artificial Neural Network; Ensemble Modelling; Feature Importance.

I. INTRODUCTION

Disk failures are not rare in Datacenter and cloud computing environments. Fortunately, we have S.M.A.R.T. (Self-Monitoring, Analysis, and Reporting Technology; often written as SMART) logs collected from computer hard disk drives (HDDs), Solid-State Drives (SSDs) and eMMC drives that detects and reports on various indicators of drive reliability, with the intent of enabling the anticipation of hardware failures. Hence, HDD vendors are highly motivated to reduce the rate of failures as a cost saving measure. SMART attributes represent HDD health statistics such as the number of scan errors, real location counts and probational counts of a HDD. If a certain attribute considered critical to HDD health goes above its threshold value, the HDD is marked as likely to fail. This report focuses on applying machine learning to improve prediction accuracy over heuristics in hard disk drives.

II. DATA DESCRIPTION

A. Data Source

The dataset under consideration is hard drive dataset, published by Backblaze. Backblaze records SMART stats of 67,814 hard drives, which are running every day in their Sacramento data center. SMART stands for Self-Monitoring, Analysis and Reporting Technology, is a monitoring system included in hard drives to report attributes about a given drive. The data for the following experiment in a cleaner format can be found on Kaggle:
B. **Data Summary**

Each day a snapshot of each operational hard drive is taken in the Backblaze data center. The snapshot will have the basic drive information along with the SMART statistics reported by that drive. The daily snapshot of one drive is one record or row of data. The snapshots of the drives are compiled into a single file which further consists of separate row to which denotes the status of the hard drive. The detailed description of dataset is as follows.

- **Date** – The date of the file in yyyy-mm-dd format.
- **Serial Number** – The manufacturer-assigned serial number of the drive.
- **Model** – The manufacturer-assigned model number of the drive.
- **Capacity** – The drive capacity in bytes.
- **Failure** – Contains a “0” if the drive is OK. Contains a “1” if this is the last day the drive was operational before failing.
- **SMART Stats** – 95 columns of data, that are the Raw and Normalized values for 45 different SMART stats as reported by the given drive. Each value is the number reported by the drive.

III. **EXPLORATORY DATA ANALYSIS**

The full version of the dataset comprises of day-wise observations covering ~ 67,814 hard drives over the span of Jan 2015 – Dec 2017. We consider taking Seagate model number ST4000DM000 as a subset for the analysis. This subset shall be henceforth referred to simply as the dataset in this document. The following is the correlation matrix generated using the SMART raw values: [Figure 3: Pearson - Correlation Matrix] Inferences drawn from the correlation matrix are as follows:

- SMART 4 and 192 exhibit high correlation as they relate to the number of cycles on start after shutdown. 192 captures power off cycles and is complemented by 4 which increments the value on startup.
- SMART 190 and 194 deal with temperature, hence highly correlated.
- SMART 197 and 198 exhibit high correlation because 197 defines unstable sectors due to read errors and 198 gives count of uncorrectable errors while read/write to a sector. We take 198 and ignore 197.
- SMART 9,12 are correlated to an extent as they cover related features - number of hours the drive is up, count of full power on/off cycles, and the Logical Block Addresses read during the time it was up.
IV. MODELLING

A. Selecting the Model

The following modeling techniques were used to predict the hard disk failure using the extracted features:

- Random Forest
- Fully connected Artificial Neural Network
- XGboost

Few features we considered before using the listed models:

In our given problem it is easy to interpret and straightforward to visualize, this will help us to explain it to business. - One important caveat in using the decision tree model is that it tries to make an optimal prediction at every node level. This makes it prone to overfitting, especially when it is deep. This is due to the amount of specificity at each node level. To avoid this pitfall, we build a random forest to compare with it. - After a thorough cleaning and applying principal component analysis, the data set was set up for Artificial Neural Network and Extreme Boosting Algorithm(XgBoost) as they performed well even on a very imbalanced anomaly detection.

B. Building the Model

The dataset was split into train and test sets based on the test design. For each trial, the records were sampled using stratified random sampling and trained on each of the selected model. 5-fold cross validation was the evaluation criteria before the final test prediction. PCA was used as the dimensionality reduction technique and the Principal Components were generated for each validation set using the coefficients...
in each of the cross-validation fold. The same approach was also considered for the final test sets. The experiment was setup on Jupyter Lab with GPU support. Also used Google Colaboratory using Keras Library on Tensorflow backend.

The [Figure 2: Feature Importance] depicted by Running XGBoost estimator, the following features were the only ones listed (in order of importance)

Smart 9, Smart 1, Smart 197, Smart 187, Smart 4, Smart 193, Smart 198, Smart 190, Smart 189, Smart 5, Smart 12, Smart 199, Smart 194, Smart 192, Smart 183, Smart 184.

Depending on the type of output the model creates we assess them differently. As assessing a model is an integral part of model building these parameters were taken into consideration:

Confusion Matrix: It is an N*N matrix, where N denotes no of classes predicted. From the confusion matrix we derive these sub-metrics:

Accuracy - % of correct predictions

Kappa - a metric that compares an Observed Accuracy with an Expected Accuracy

Precision - % of positive cases which were correct

Sensitivity/Recall - a portion of actual positive cases which were predicted correctly Specificity - a portion of actual negative cases which were predicted correctly.

ROC and Area under the Curve - A plot of TPR vs FPR to showcase the strength of the model.

V. MODEL PERFORMANCE
All the model evaluation metrics are shown in [Figure 3: Model Performance Evaluation]. One of the most important metric for the experiment being Fmeasure - % of correct predictions from for Type II error.

<table>
<thead>
<tr>
<th>True condition</th>
<th>Condition positive</th>
<th>Condition negative</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total population</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Predicted condition</strong></td>
<td>True positive, Power</td>
<td>False negative, Type II error</td>
</tr>
<tr>
<td><strong>False negative, Type II error</strong></td>
<td>True negative</td>
<td></td>
</tr>
</tbody>
</table>

The models trained on the new feature set gave better results than the models trained on just last day's data. Amongst the three models trained the XGboost and Neural Network models gave the most optimal accuracy and sensitivity. But considering the time needed to train the model, the bagging based algorithm was faster to train.

VI. CONCLUSION

In this experiment, we have analysed the Backblaze hard drive (Seagate - ST4000DM000) failure and used several prediction models for classification. We evaluated the prediction performance among Random Forest, Gradient Boosting (XGBoost) and Artificial Neural Network (ANN) models. The data set was highly imbalance in nature also the SMART statistics alone cannot provide the best model as the other constraints for hard disk failure are numerous and not all constraints can be monitored. The false negative rate was high due to those failed hard drives not having any relevant SMART data indicators for failure.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total number of drives</strong></td>
<td>11,569</td>
</tr>
<tr>
<td><strong>Number of failed drives</strong></td>
<td>42</td>
</tr>
<tr>
<td><strong>Number of predicted failures</strong></td>
<td>25</td>
</tr>
<tr>
<td><strong>Number of false positives</strong></td>
<td>9</td>
</tr>
<tr>
<td><strong>Number of false negatives</strong></td>
<td>26</td>
</tr>
<tr>
<td><strong>Number of true negatives</strong></td>
<td>0</td>
</tr>
<tr>
<td><strong>Number of true positives</strong></td>
<td>16</td>
</tr>
<tr>
<td><strong>Fmeasure</strong></td>
<td>0.733944954128</td>
</tr>
</tbody>
</table>

In this report we have limited our scope to only Seagate model number ST4000DM000, our analysis and prediction can be further extended to other hard drive models from Backblaze. Furthermore, we are now predicting the hard drive device failure on the day of its failure. If we could predict the device failure in advance, then suitable backup action can be taken to avoid the data loss. Training time of Fully connected Neural Network can be reduced by building efficient RNN and LSTM networks that can also self-analyse and predict the hard drive failure in advance.