

Failure Prediction, Lead Time Estimation and Health Degree Assessment for Hard Disk Drives Using Voting Based Decision Trees

Kamaljit Kaur^{1,*} and Kuljit Kaur²

Abstract: Hard Disk drives (HDDs) are an essential component of cloud computing and big data, responsible for storing humongous volumes of collected data. However, HDD failures pose a huge challenge to big data servers and cloud service providers. Every year, about 10% disk drives used in servers crash at least twice, lead to data loss, recovery cost and lower reliability. Recently, the researchers have used SMART parameters to develop various prediction techniques, however, these methods need to be improved for reliability and real-world usage due to the following factors: they lack the ability to consider the gradual change/deterioration of HDDs; they have failed to handle data unbalancing and biases problem; they don't have adequate mechanisms for health status prediction of HDDs. This paper introduces a novel voting-based decision tree classifier to cater failure prediction, a balance splitting algorithm for the data unbalancing problem, an advanced procedure for lead time estimation and R-CNN based approach for health status estimation. Our system works robustly by considering a gradual change in SMART parameters. The system is rigorously tested on 3 datasets and it delivered benchmarks results as compared to the state of the art.

Keywords: Hard disk drive, lead time, health status, N-splitting algorithm, machine learning, deep learning, data storage, unbalancing problem.

1 Introduction

The 21st century is the era of big data, cloud computing, and distributed networks. Either the case is, data storage, data reliability, and data accessibility are becoming big challenge day by day. Tremendous upsurge in data exchange and resources has led to compensate the need for sophisticated, reliable and affordable storage systems. A formidable technique that has been used for storage is the Hard Disk Drives (HDDs). This raises a concern to ensure that these HDDs are at optimal functionality and high operation level to enable the smooth running of processes at data centers.

Bulky storage devices (HDDs) are extensively exploited in high-performance computers, cloud data centers, back-up storage, extra storage memory, Internet Service Providers (ISPs) and data synchronization. The enormous volume of data, present in these settings,

¹Kamaljit Kaur, Department of Computer Engg & Technology, Guru Nanak Dev University, Amritsar, Punjab, 143005, India.

² Kuljit Kaur, Department. of Computer Science, Guru Nanak Dev University, Amritsar, Punjab, 143005, India.

*Corresponding Author: Kamaljit Kaur. Email: kamal.aujla86@gmail.com.

enlarges the chances of failures in hard drives. Vishwanath and Nagappan described that HDDs are amongst the top reasons for the failure of data centers [Vishwanath and Nagappan (2010)]. Which translates that in big datacenters, hard drives fail once per day [Xin, Miller, Schwarz et al. (2003)]. According to Schroeder and Gibson, the failure rate of HDDs has been exceeded by 10% annually [Schroeder and Gibson (2007)].

HDDs failure can lead to catastrophic consequences which can be unrecoverable and permanent. This can cost in the form of server down, erased backup, lower reliability, unavailability of Internet and fiasco to fetch the latest data and compromised data storage in the data centers. To minimize the problem, HDDs conditions are monitored which helps in the detection of soon to fail drives [Strom, Lee, Tyndall et al. (2007); Ma, Traylor, Douglass et al. (2015); Yang, Hu, Liu et al. (2015)]. The conditions of HDDs are supervised using sensors like acoustic emission, accelerometers, counters and thermal sensors. However, the results provided by these techniques are not accurate enough in real time [Pecht, Tuchband, Vichare et al. (2007)]. Moreover, many algorithms have proposed efficient prediction and inspection of the health degree of HDDs. In most cases, various approaches involving artificial intelligence (AI) and mathematical computations have been applied. However, these methods have three types of shortcomings.

First, the problem that arises due to uniform classification [Mak, Phongtharapat and Suchatpong (2014)]. There are several categories of failure that can be experienced by an HDD. This occurrence is attributed to the sophisticated structure of the disk drives. Frequent failure is therefore easily determined as opposed to failure that occurs less often -Thus, making the uniform classification method less efficient. Secondly, there is a problem brought about by data unbalance, as a result of a huge difference between failed drives and good drives [Longadge and Dongre (2013)]. Unbalanced data can reduce the efficiency of HDD failure assessment and lead time prediction. The third problem is that only a few researchers have worked on lead time prediction method for HDD health assessment [Salfner, Lenk and Malek (2010)].

Few researchers tried to resolve these problems using proactive approaches, where the failure of HDDs can be predicted before it really occurs. To gain efficient results, numerous machine learning, statistical and data science approaches have been employed. These methods work on the principle of SMART (Self-Monitoring Analysis and Reporting Technology) features. Despite the fact that these methods are extremely effective, they have limitations as well. For example, these approaches only produce a binary classification about the health of HDDs i.e. whether good or bad. Similarly, they cannot differentiate if HDDs are close to failure or have some time before failure. Furthermore, these methods rely on straightforward SMART features without considering the different health statuses in multiple time zones [Zhu, Wang, Liu et al. (2013)]. The mentioned shortcomings have led us to design a robust methodology by the application of SMART attributes to get the concise synopsis of the health statuses of various hard drives. Our main objective is to provide an efficient technique to predict failure of HDDs by using the voting-based decision tree model. This model functions by establishing a number of decision trees for classification. The various combinations of decision trees provide the advantage of being able to differentiate between good drives and faulty drives. Each tree focuses on classifying a section of health parts rather than all of them. The issue

of data unbalancing is solved by partitioning the dataset into n -splits. While decision trees are used on each sub-group in a parallel manner. For training purpose total n -classifications (Binary) are done by n -decision trees, on the results of n -classifications, voting is applied. If more than $N/2$ voters have predicted the negative result, then HDD is classified to be failed. This concept of $N/2$ voters is derived from Li et al. [Li, Ji, Jia et al. (2014)]. When the HDD is predicted to be failed, its lead time (Time Before Failure) is predicted and health degree is examined. So, our proposed work can help engineers to save huge volumes of data and provide lead time to perform data transfers before HDDs actually fail.

This article focuses on the following important issues:

- We have proposed a novel failure prediction technique for HDDs. It lays a basis on voting-based decision tree model;
- Dealing with the problem of data unbalancing by grouping data into n -subsets – Thus, enhancing the quality of training data set;
- Lead-time prediction and health degree assessment of HDDs;
- Detailed analysis using data from production centers, which indicate that prediction models can obtain an FDR above 99.99% and an FAR of 0.001%;

The remaining sections of the paper are organized in the following manner. Section II focuses on recent works on HDD and its failure prediction by utilizing SMART technology. Section III defines the prediction models and associated step by step procedures. Section IV talks on for failure prediction estimation. Section V lays the basis of the experiments and related findings. Section VI summarizes the entire article with some discussions.

2 Background and recent works

2.1 Hard disk drive failure

Due to the powerful mechanism, extensive usage and complex structure of the drives, HDD failures can be differentiated into different categories [Huang, Fu, Zhang et al. (2015)]. The structure of HDD can be divided into various components: platters, the substrate material, read & write heads, the spindle motor, HDDs internal logic board and drive bay [Wang, Miao and Pecht (2011)]. The working of each component is unique, which is why the trigger factors for component failure are distinct. For instance, the mechanical parts like read/write heads, spindle motor and drive bay often writhe from mechanical failure. Whereas, logic board and substrate material often disturbed by short circuits. Mechanical failure of components is easier to identify and predict long before while, the short circuits are complex to identify and hard to predict.

There is relatively less published literature on the failure patterns of hard disk drives. A variety of failure prevailing stresses is humidity, temperature, CSS (contact-start-stop) frequency, altitude and duty cycles [Xu, Wang, Liu et al. (2016)]. Due to increase in the temperature that may be caused by an increase in the workload of the hard disk drive or of the whole system, may corrupt some sectors of the drive, labeled as latent sector errors [Bairavasundaram, Goodson, Pasupathy et al. (2007)] and silent data corruption [Bairavasundaram, Arpaci-Dusseau, Arpaci-Dusseau et al. (2008)].

A plethora of research has been conducted on HDD failure classification. Identification of the HDD failure with respect to the mechanism, mode, and cause is given by Wang et al. [Wang, Miao and Pecht (2011)]. Similarly, the classification of HDD failure into three subclasses (bad sector; logical; read/write head) is provided by Huang et al. [Huang, Fu, Zhang et al. (2015)]. Moreover, they proposed that the classification of HDD failure can help to make the HDDs failure prediction mechanism robust. After extensive experiments, it is found that the straightforward classification is not accurate enough for failure prediction. To enhance the power of failure prediction systems, usage of SMART parameters has become state of the art recently [Zimmer and Rothman (2007)].

2.2 SMART (Self-monitoring analysis and reporting technology)

SMART is an in-built HDD function, that works by calculating attribute's values, used to evaluate the performance of HDDs [Murray, Hughes and Kreutz (2005); Wang, Miao and Pecht (2011)]. This is achieved by a series of record counts by sensors and counters. The forthcoming data from SMART attributes is very detailed as it contains more than 30 drive features. These include temperature Celsius (TC), reallocated sector count (RSC), power-on-hours (POH), seek error rate (SER), and the spin-up time (SUT). The purpose of these variant attributes is to show the health status of HDDs.

Each of these attributes is composed of five fields. These include raw data, threshold, value, worst case, and the status. Raw data values are recorded by the sensor. Value is the standardized measure of raw data. The threshold is essential in failure detection. The status is responsible for giving a warning any time; the standardized value exceeds the value of raw data. The value of the threshold is vendor specific and usually fixed by the HDD manufacturers. When the threshold value exceeds the normalized value, a warning is triggered. However, HDD manufacturers have analyzed that threshold-based methods can detect failure-rate of 3-10% only [Murray, Org, Hughes et al. (2005)].

SMART attributes are strongly correlated with HDD failure. The implication of 3-SMART parameters and all-SMART parameters for the prediction of HDD failure is tested and concluded with the results that all-SMART parameters give significant failure detection rate [Hamerly, Elkan, Diego et al. (2001)]. The RSC (reported scan error) is tested and strongly correlated with the HDDs failure [Pinheiro, Weber and Barroso (2007)]. Various studies and research work have shown that these attributes are in close relation to the failure of HDDs. There are various categories for the SMART-based failure prediction methods which exist today. These include threshold-based methods, statistical methods, binary status prediction of hard drives and the lead time prediction.

It was suggested that the threshold-based technique is ineffective when employed to enlarge an HDD's mean time to data loss (MTTDL) [Eckart, Chen, He et al. (2008)]. Experimentally, when thresholds were kept low, this method provided low accuracy in prediction rate which is unacceptable. Well, when the thresholds were kept high, SMART produced extreme false positives. They have argued that reactive failure prediction methods are helpful in extension of overall MTTDL. They achieved a 50% sensitivity on the dataset (consists of a time series of SMART attributes of a single drive model) [Murray, Hughes and Kreutz (2005)], by combining SVM with proactive fault tolerance approaches.

It has been identified that, numerous parameters from SMART are highly correlated with HDDs failure. They conducted vast experiments and concluded that only SMART attributes are not enough for failure prediction. Moreover, the usage of HDDs and room temperature are least correlated factors in HDDs failure cases [Pineiro, Weber and Barroso (2007)].

In Hughes et al. [Hughes, Murray, Kreutz et al. (2002)], improved SMART algorithms were proposed. They exploited the maximum error threshold warning-algorithm by replacing it with statistical hypothesis assessments (distribution-free). These warning algorithms can be easily replaced and implemented in HDDs because of low computation cost. A total of 3744 drives of 2 models were tested by these algorithms. They achieved 3 times better prediction accuracy with only 0.2% false-alarm rate. These algorithms are not highly employed because they require a high level of care and only provide 40% accuracy [Hughes, Murray, Kreutz et al. (2002)].

In Li et al. [Li, Ji, Jia et al. (2014)], a novel HDDs failure prediction model is implemented. Their methodology is based on regression trees and classification trees which is robust in the prediction performance, interpretability and stability while comparing with state-of-the-art models. Experiments validated that their model predicts more than 95% of accuracy and under 0.1% false detection on a real-world dataset of HDDs.

In Murray et al. [Murray, Hughes and Kreutz (2005)], an efficient comparison of prediction algorithms are provided. They have proposed two major contributions: First, they divided the prediction problem into a multilevel classification problem and implemented a novel algorithm by combining multiple instances with naive Bayes (mi-NB); Secondly, they highlighted computational efficiency, usage, and effectiveness of non-parametric statistical methods while comparing them to state of the art learning methods.

The performance of the total 21 machine learning techniques was evaluated for HDDs failure prediction problem. They exploited the power of WEKA for experimentation and test all the publicly available benchmarks that were used for HDDs failure prediction. They stated the results in terms of different constraints and explained that each ML technique has significant advantages with few limitations [Pitakrat, Van Hoorn and Grunske (2013)].

The paradigm of anomaly detection for the failure forecasting of HDDs was exploited. Their system uses non-parametric and semi-parametric techniques. Primarily, SMART attributes are collected from safe HDDs and a Gaussian mixture model (GMM) is designed. Furthermore, the SMART attributes are compared with the GMM model' output and dissimilarity vector are generated in a time interval T. Finally, the anomaly is perceived if the threshold of Kullback-Leibler divergence (KLD) surpasses a limit [Queiroz, Rodrigues, Gomes et al. (2016)].

The task of discarding the SMART attributes was performed which were not present in 90% of disks and choose 21 contributing SMART attributes. The reverse Arrangement test was applied to get the attributes which are more likely to be used as disk failure parameters. They trained a decision tree that recognized true parameters which are responsible for HDDs failure. They employed Backblaze data to assess the correctness of method and stated 52% accuracy [Rincon, Paris, Vilalta et al. (2017)].

The capabilities of different Bayesian techniques were investigated based on HDDs internal conditions. This is helpful in the prediction task of HDDs failure. Moreover, they introduced a novel model which is trained by exploiting the power of expectation-maximization. They also introduced a naive Bayes based classifier. Experiments have been conducted on real-world data containing a total of 1936 drives. The accuracy of both techniques is comparable with state-of-the-art methods [Hamerly and Elkan (2001)].

Part-voting random forest technique was proposed for the HDDs failure prediction. Their methodology is novel and has a less false detection rate. They have conducted a vast amount of validation experiments on real-world datasets having the 64,193 HDDs data (SMART parameters). They have achieved a 5% better prediction accuracy than already present methods [Shen, Wan, Lim et al. (2018)].

A novel method for HDDs failure prediction was developed by using Mahalanobis distance (MD). They selected parameters which were highly reasonable for failure in different settings. Testing is employed on the SMART data set. The method provided 67% detection rate with zero false alarm generation. It can provide more than 20 hours prediction time, which can be useful to generate backup [Wang, Miao, Ma et al. (2013)].

The best detection rate was reported by the implication of anomaly detection principle with zero false alarm rate. The method used Mahalanobis distance and Box-Cox transformation to model the statistical behaviour of healthy or good HDDs for estimating the deterioration process of an HDD. They have used the GLR test rather than dissimilar vector approach for the detection of faults. A cost function has also been implemented for the reduction of false alarms to 0% [Wang, Ma, Chow et al. (2013)].

A novel method inspired by Recurrent Neural Networks (RNN) was introduced to examine the health condition of HDDs. They altered the consecutive attributes of SMART steadily. Having the knowledge of HDD's health condition pays a lot in contrast to simple prediction. Health condition can provide the level of urgency so that engineers can schedule the recovery process. They conducted experiments on real datasets and demonstrated comparable results in health status measuring [Xu, Wang, Liu et al. (2016)].

Despite the fact that a lot of research contribution is present for HDDs failure predictions, the results are not up to the mark. Due to high false detection rate, these methods are not deployable in the real-world environment. Some methods only provide a review of HDDs failure classifications which is not useful for implementation in real-world data centers, other methods use highly unbalanced data, which is small, old and biased. The accuracy is not good enough and the false detection rate is high. To solve these issues, we have proposed a state-of-the-art method, using data balancing techniques to avoid model overfitting and false detection issues. This can be achieved by using our proposed voting based decision tree method. Once the HDDs are predicted to be failed, we have analyzed its lead time (the time we have before it actually fails). Moreover, the health degree of the HDD is predicted. Finally, testing is conducted on the real-world data set, having more than 75 thousand HDDs data and achieved comparable results. The details of the implementation of the proposed methodology are provided in the following section.

3 Proposed methodology

3.1 HDDs data normalization

Data normalization is a fair way to compare different features by reducing the scaling effects. Normalization techniques are widely exploited in machine learning, data processing, and statistical analysis. The SMRR dataset was already normalized, while Backblaze dataset is normalized using flowing formula:

$$feature_n = \frac{feature - feature_{min}}{feature_{max} - feature_{min}} \quad (1)$$

where a feature is an original value from the dataset, $feature_{max}$ and $feature_{min}$ are minimum and maximum values of this feature.

3.2 Feature selection based on RF-RFE

For feature selection, a robust features elimination technique is adopted from Granitto et al. [Granitto, Furlanello, Biasioli et al. (2006)]. This is a recursive technique which provides the ranks of a feature by measuring its importance. Algorithm 1 illustrates the pseudo-code for RFE. Technically, multiple iterations are performed and the importance is ranked at the end of each iteration while the irrelevant features are eliminated. Usage of recursion here is necessary because the comparative rank of a feature may get altered if evaluated with a different subclass in the stepwise exclusion procedure. The features are eliminated in a reverse manner (recursion) and the final ranking is generated. Finally, a feature selection method provides the n features having K ranking with respect to their importance.

Algorithm 1: Feature selection technique using RFE.

Inputs:

Training set T_s

Feature set having n number of features $F_s = \{f_1, f_2, f_3, \dots, f_n\}$

Technique for ranking $Fr (T_s, F_s)$

Outputs:

Final ranking K

Begin:

Foreach value of m

Rank set K using $Fr (T_s, F_s)$

$f^* \leftarrow$ last ranked feature in F

$R(m - i + 1) \leftarrow f^*$

$F \leftarrow F - f^*$

End For

End.

The recursive feature elimination (RFE) is further combined with random forest (RF). Due to the fact that tree can be grown only on bootstrap, each tree of RF has out-of-bag (OOB) subset taken from learning set. The OOB set never got exploited during a training session. The measurement of unbiased classification error can be computed by using

OOB. When a feature enters to the model, its relevance can be measured in the following manners: Each feature got shuffled (one at a time) and its prediction error is computed using OOB set, for the ‘shuffled’ data set. Naturally, the prediction error for irrelevant features will remain the same using this method, while the relevant feature’s prediction error will keep minimizing at each iteration. Technically, RF-RFE method efficiently measures the importance of each feature without employing extra computation cost.

Furthermore, we employed a nonparametric method of Kruskal Wallis Statistic (KWS) for the computation of generalized ranking. Because of the fact that KWS is a single variable measurement, its rank does not get altered for every individual feature, while evaluating the different subsets. Which means that recursion turns out to be irrelevant, and univariate calculation is required. So, we have straightforwardly computed the KWS for every feature and produced ranks. The following Tab. 1 provides the ranks of Backblaze dataset and Tab. 2 illustrates the ranks for Z family data set.

Table 1: Feature importance along with their ranks for backblaze dataset [Beach (2014)]

SMART Attribute	Description	Rank
smart_9_raw	Power-On Hours Count	53.0001392097
smart_7_raw	Seek Error Rate	12.9280721442
smart_189_raw	High Fly Writes	10.4504652433
smart_194_raw	Temperature	9.53915152317
smart_1_raw	Read Error Rate	6.65939348074
smart_1_raw_diff	Read Error Rate	3.74839267095
smart_187_raw	Reported Incorrect able Errors	3.40816126783
smart_197_raw	Current Pending Sectors	2.09317627002
smart_194_raw_diff	Temperature	1.62027737629
smart_5_raw	Reallocated Sectors Count	0.898169866234
smart_198_raw	Off-line Incorrect able	0.759965782582
smart_9_raw_diff	Power-On Hours Count	0.583214055371
smart_188_raw	Command Timeout	0.263305212294
smart_7_raw_diff	Seek Error Rate	0.0133497558702
smart_187_raw_diff	Reported Incorrect able Errors	0.0075442541443
smart_189_raw_diff	High Fly Writes	0.00444313837005

As we can see from the Tab. 1, in Blackblaze data set; “smart_9_raw” attribute, got the highest rank (it is the most correlated with failure prediction), while “smart_187_raw_diff”, have the lowest rank in the given set of highly-correlated SMART attributes. Similarly, from Z family dataset; attribute “Servo5” got the highest rank and attribute “Reads” got the lowest rank in top correlated SMART features.

Table 2: Feature importance along with their ranks for Z family dataset [Murray, Hughes and Kreutz (2005)]

SMART ID	Feature Rank
Servo5	14.4943469318
Temp4	13.2249458146
Temp1	11.0276073094
CSS	9.63347255597
Servo10	7.98472891709
Writes	6.51517745317
Temp3	5.87563577037
FlyHeight11	3.09877286055
Servo9	2.54603177138
Servo6	2.18108705889
FlyHeight6	1.99930382918
Reads	1.884123018

3.3 Balanced splitter algorithm (BSA) for unbalancing problem

As we have already described in the previous section, that the available data for HDDs failure is either too much normalized or highly unbalanced. The problem with these methods relying heavily on normalized data is that they provide efficient training accuracy, but in real time these methods failed to provide reliable results. So, using normalized data sets for training is inefficient and promotes the wastage of resources. Similarly, the usage of the unbalanced dataset is ineffective. Unbalanced means, there is fewer number of bad samples as compared to good samples. If the HDD is kept plugged in the storage and still working it is considered as a good SMART sample, on the other hand, if an HDD is supplanted due to the occurrence of any failure, it is considered as a bad SMART sample.

The good samples can be recorded any time, but samples of bad drives are quite less. So, unbalancing of data can cause biases and the resultant model will have compromised accuracy. The most utilized method for this unbalancing problem of data is under-sampling [Yen and Lee (2009)], which works by randomly choosing a data sample from the healthy set and merging it with the corrupt HDDs for training purpose only. But there are a number of problems with the under-sampling method as follows:

- The data can be highly biased because of random selection, (i.e. only one type of HDDs got selected);
- Thus, reduced the amount of training data; lesser the training data lower the accuracy of model [Khan, Farooq, Hussain et al. (2019)];
- Few random samples are chosen from good HDDs, what is the goodness factor of those samples;
- Random selection of good samples for training may cause over-fitting of the model;

To resolve these issues, we have proposed a novel algorithm, called Balanced Splitter Algorithm (BSA), which divide all the good samples into n -subsets (based on the number of bad samples). BSA helps to train the model by utilizing each combination of the good sample with the available bad samples. Algorithm 2 explains the detailed procedure of BSA.

Algorithm 2: *Creating balanced data for Training*

Input:

Dataset having SMART samples of the HDDs;

Output:

n -splits of SMART samples for each HDD;

Begin:

For each hard disk drive in Dataset Do

Identify no. of good samples

Identify no. of bad samples

n = good samples' ratio with respect to bad samples

balance the data by dividing all samples into n -sub-groups of the same size by using a combination of all the good samples and bad samples.

End For

End

This algorithm provides the resourceful combinations of the complete dataset without missing any good samples. It also enhances the efficiency of the model because of using a large volume of data for training the model. After successfully solving data unbalancing problem, the next step is to build a model for failure prediction. The following section will elaborate on the failure prediction mechanism of HDDs.

3.4 Decision tree model for classification based on GINI index

The measurement of inequality of distribution (the ratio of value from 0-1) is called the Gini coefficient (G_iC) [Gastwirth (1972)] Technically, it is the ratio of following measures: the area (Lorenz curve to the uniform line of distribution); the area of the uniform line of distribution. If the G_iC is articulated as a percentage it is called Gini index. This calculation and relation of G_iC and Gini index can be derived from as Eqs. (2) and (3)

$$G_iC = \frac{\sum_{j=1}^n \sum_{k=1}^n |x_j - x_k|}{2 \sum_{j=1}^n \sum_{k=1}^n x_k} \quad (2)$$

$$Gini\ index = G_iC \times 100 \quad (3)$$

where x_j here is Lorenz curve of the distribution, while x_k is a uniform line of distribution. Decision trees are built by exploiting GINI index-based splitting, which employs a combination of tree structure and voting mechanism to build the classification model. It divides SMART attributes to make small subsets by using GINI index based splitting function. The final outcome is the decision tree with a classification output.

To generate classification trees recursive technique is exploited for partitioning. The nodes are split until the split criteria, max depth, or max leaf nodes are completed. It can

also stop when it has only a single class. After the generation of classification trees for n subsets of all samples, a voting-based method is used for final prediction. If more than $N/2$ voters have positive classification, HDD will be predicted as failed. The explanation of voting method is provided in Algorithm 3. While the complete architecture diagram of HDDs failure prediction is explained in Fig. 1.

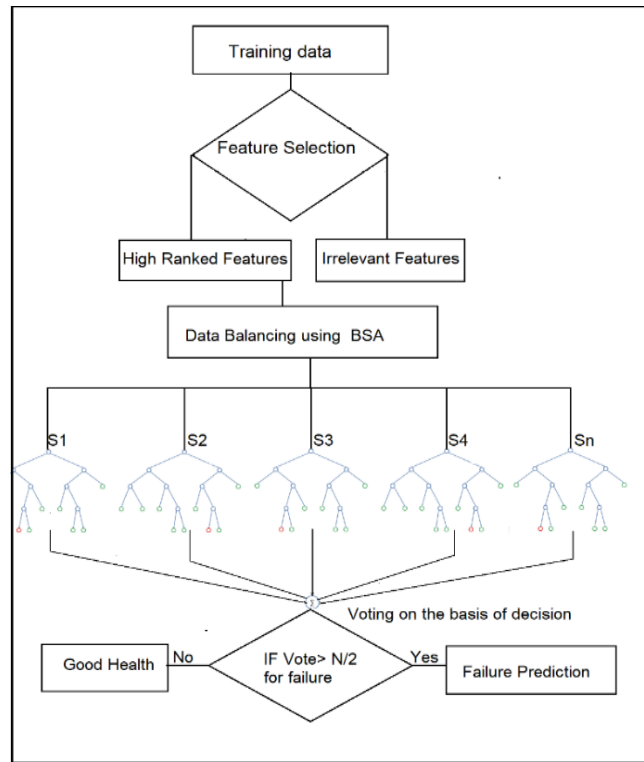


Figure 1: Detailed architecture of the proposed methodology

Algorithm 3: Voting based decision tree model for failure prediction.

Input:

n -sub-group of SMART samples for training the model, n - decision tree classifier, Split criterion, max-depth, max-features, max-leaf-nodes, splitter.

Output:

The prediction results.

Begin:

For each sub-group of SMART samples.

Call decision tree classifier, apply voting on classification results of all decision trees.

If more than $n/2$ voters vote failure prediction

Then HDD is about to fail

Else HDD is good
End if

End for
End.

3.5 Lead time prediction

The failure prediction of HDDs is not enough, as engineers required the time before it actually fails (lead time). So, they can initiate the backup of data in a timely manner. The lead time of HDDs is predicted using some relevant parameters. These parameters are used by Algorithm 4 for accurate lead time prediction. The parameters are described below:

- First day: The initial day of disk monitoring by utilizing SMART parameters;
- Length: The total time from the first day till the last day when the disk kept working;
- Predicted day: The day when the drive is predicted to be failed;

Algorithm 4: *Lead Time prediction using hyperparameters.*

Input:

First day, Length, Predicted day/hour

Output:

The Predicted Lead time for HDD.

Begin:

For each disk drive in D Do

IF HDD is predicted to be failed by algorithm 3

Predicted day = day of the year

Or

Predicted hour = hour of the day

Lead time= length - predicted day

Or

Lead time = length - predicted hour

End IF

End for

End.

By exploiting the above-mentioned algorithm, we have predicted lead time starting with 10 days till more than 90 days for Backblaze dataset. According to testing, after prediction of failure more than half of the HDDs may survive up to 90 days. A small portion of HDDs showed a critical condition (about to fail in 10 or fewer days). Similarly, we predicted the lead time for hourly dataset ranging from 24 hours to 500 hours. In this regard, our system can timely alarm for data backup and other precautionary measurements. The algorithm4 is exploited for the prediction of lead time by utilizing the mentioned parameters.

3.6 Health status prediction

Apart from lead time prediction, we have also examined the health status of HDDs and further subdivided this status into six sub-classes with respect to their health status. The health status forecasting technique can significantly expand the reliability of HDDs. In the way, Engineers could easily schedule the backup procedure and effectively manage the priorities of backup according to HDD's health status. Conclusively, the probability of data crashing can be reduced in an effective manner and resource allocation can be maintained accordingly, which can save huge data loss and increase resource optimization.

Fig. 2 Explains the possible 6 classes for the hourly dataset (SMRR). Level 6 here is a safe zone, while level 1 means that the drive is in critical condition and its data needed to be backed up. The reason to choose 6 classes for health status prediction is borrowed from Xu et al. [Xu, Wang, Liu et al. (2016)]. Experiments have proven that the health status of HDDs changes slowly in a monotonic way, which can be observed by constantly monitoring SMART attributes for a long time period. For example, if the disk is used frequently in the shorter period the attribute, Temperature Celsius (TC) can change in an enormous way. This kind of change in an attribute can confuse the prediction method. These kinds of confusing attributes should be measured over a long specific period of time.

This dependency can be measured as the high-degree property of Markov: which means that the current state has a stronger correlation than previous states. To measure Markov value the key is to evaluate the conditional entropy of features. Let T be the time series data of smart attributes ranging from 1 to n , $\{T^1, T^2, T^3, \dots, T^n\}$, the entropy can be defined as (4).

$$H(Y|Z) = - \sum_{q \in X} \text{prob}(q) \sum_{j=0}^{N_{lbl}} \text{prob}(hlt_j|q) \log \text{prob}(y_j|q) \quad (4)$$

where hlt is health degree, N_{lbl} are the total number for health labels, for order $n-1$ feature representation, $q = \{T^t, T^{t-1}, \dots, T^{t-n+1}\}$. Extensive experiments have discovered that keeping the value of $N_{lbl} = 6$, provide the optimal status of health degree [Xu, Wang, Liu et al. (2016)]. That is why we have kept 6 health levels.



Figure 2: Health status classes of the hourly dataset (SMRR)

Similarly, Fig. 3 illustrates the possible classes for health degree prediction. In this setting, 6 possible classes of health degree are proposed as follows:

- Level 6 specifies that HDD is working accurately;
- Level 5 explains that HDD has fair health status, but it needed to be monitored and rechecked again after some time;
- Levels 1-4 signifies that HDD will fail;
- Level 1 provides alert that HDD has less than 10 days;



Figure 3: Health status classes or yearly dataset (Backblaze)

For the implementation of HDDs health degree measurement setup, we borrowed a concept from Xu et al. [Xu, Wang, Liu et al. (2016)]. The training setup uses a convolution neural network (CNN) with hidden layers. The recurrent weights R are shared between all the hidden layers. The benefit of using a recurrent weight is that the hidden layers have local information alongside with global information by keeping the sequential information of previous weights. Moreover, there are a few chances of vanishing gradient problem in the recurrent neural network. Stochastic Gradient Descent (SGD) is exploited for optimization of RCNN. The gradient can be calculated as (5)

$$Err_0(T) = actual(T) - health(T) \quad (5)$$

where $health(T)$ is predicted health degree, and $actual(T)$ is the actual health degree. The weights δ , from hidden layers $hid(T)$ to output unit $health(T)$ are regularized as (6)

$$\delta(T+1) = \delta(T) + hid(T)Err_0(T)^T\alpha - \delta(T)\lambda \quad (6)$$

α here is used as hyperparameter for learning rate while λ is weight decay parameter. Similarly, $Err_0(T)^T$ is the resultant of transposition of SGD function. The recurrent weight R updated as (5)

$$R(T+1) = R(T) + hid(T-1)Err_{hid}(T)^T\alpha - R(T)\lambda \quad (7)$$

where Err_{hid} is loss of hidden layer and $Err_{hid}(k)^T$ is the transpose of Err_{hid} .

4 Dataset

The dataset we have used for failure prediction is gathered from two sources as follows: the first main source of the dataset is Backblaze data center [Beach (2014)] and the second source is Center for Magnetic Recording Research [Murray, Hughes and Kreutz (2005)]. The first part of the dataset is collected in 2 years' time period, it has total 75,428 HDDs from 80 different manufacturers. For training purpose only, we choose to move forward by 2 models (ST3000DM001, ST4000DM000), having a larger count of HDDs than others. For the rest of the paper, we will describe these models as X and Y (for simplicity purpose), the dataset from the second part will be referred to as Z.

“X” family have total 2892 HDDs in good health and 1344 failed HDDs. Similarly, “Y” family have total 35,521 HDDs in good health and 1041 failed HDDs. For “X” family, there are 75,987 available samples for failed HDDs and 175,671 available samples for healthy HDDs. Similarly, for “Y” family, total 59,709 available samples are there for failed HDDs and 1,992,511 available samples for healthy HDDs. Each sample consists of 24 SMART attributes (having a value and raw data). After applying feature selection (Section III-A), 16 attributes are selected as follows:

Table 3: Selected Features for the X and Y Family (Backblaze Dataset)

SMART Attribute	SMART Attribute	SMART Attribute
smart_9_raw	smart_187_raw	smart_9_raw_diff
smart_7_raw	smart_197_raw	smart_188_raw
smart_189_raw	smart_194_raw_diff	smart_7_raw_diff
smart_194_raw	smart_5_raw	smart_194_raw
smart_1_raw_diff	smart_198_raw	smart_187_raw_diff
smart_189_raw_diff		

The second data set has time series data composed of SMART attributes. This data set has a total of 369 HDDs, having 68411 usable samples. All the drives are from the same manufacturer. From 369 HDDs, 178 HDDs in good health and 191 failed HDDs. In a semi-controlled environment, good samples of HDDs have successfully completed reliability test. While bad samples are the collection of returned HDDs. After applying feature selection, we have used the following 12 attributes of SMART, among 16 attributes, from “Z” family dataset.

Table 4: Selected Features for the Z Family (SMRR Dataset)

SMART Attribute	SMART Attribute	SMART Attribute
Servo10	Servo9	Servo5
Writes	Servo6	Temp4
Temp3	FlyHeight6	Temp1
FlyHeight11	Reads	CSS

5 Evaluation metrics

The performance of different failure prediction methods is usually evaluated by FDR, FAR, lead time, health status and accuracy. FDR can be calculated by dividing predicted near-to-fail HDDs with a total of near-to-fail HDDs as follows:

$$FDR = \frac{Num_{true\ positive}}{Num_{positive}} \quad (8)$$

As HDDs' failure prediction models consider near-to-fail drives as positive drives and good drives as negative drives. The FAR can be obtained by dividing the number of false positive with the number of negative drives (good health drives).

$$FAR = \frac{Num_{false\ positive}}{Num_{negative}} \quad (9)$$

The highest value of FDR and the lowest value of FAR makes a prediction method more reliable but this is not achievable at the same time by employing ML techniques. Therefore, the other useful method to check the correctness of a failure prediction system is accuracy evaluation. The accuracy can be derived as:

$$\text{Accuracy} = \frac{\text{Num}_{\text{true positive}} + \text{Num}_{\text{true negative}}}{\text{Num}_{\text{total}}} \quad (10)$$

Additionally, lead time and health status are also measured for evaluation. These metrics provide effective analytics of time period we have, before an HDD fails, to perform precautionary measures. This can help with data integrity and better resource management.

For the measurement of health degree Acc_{hl} is Tol_{hlt}^{acc} of health as calculated as follows:

$$Acc_{hl} = \frac{\text{preidcited health accuracy}}{\text{acutal health accuracy}} \quad (11)$$

$$Tol_{hlt}^{acc} = Acc_{hl} \mp 1_health\ level \quad (12)$$

Tol_{hlt}^{acc} provides the approximation of health status of hard drives. This is also useful in real-time scenario.

6 Experiments and results

This section will provide the details of the experimental setup that is used for the analysis of results such as Failure detection rate, false alarm rate, ROC-curve (considering various parameters- voters, time-window), lead-time, health-status, execution-time for all three families of the dataset. The final results of the evaluation are compared with the existing state-of-art methods.

6.1 Experimental setup

All the drive samples are divided into three sets, 70% Training, 15% Testing, and 15% validation. Training data is balanced by using BSA algorithm (see Section 3.3 for details). With the help of BSA, the problem of the unbalanced data is solved completely. Afterward, the model is trained to predict the failure on real-world data center, using voting-based decision tree classifier (details in Section 3.4). If a drive is predicted to be failed, its lead time is predicted using our proposed algorithm (Section 3.5). Finally, the health status of soon to failed HDD is measured in six classes by utilizing R-CNN (Section 3.6). The model is tested on random 15% data and it outperforms the state of the art. Furthermore, to ensure the performance on unseen data, the model is tested on validation data using 10-fold cross-validation and results are outstanding.

6.2 Results and evaluation

Fig. 4 shows the results of FDR and FAR of our model, when tested using a random combination of data samples and balanced samples of n-sub-groups for families X, Y and Z. Fig. 4(a) shows that, for X family, FDR ranges between 71% to 76% only, but the value of FAR varies a lot, ranges between 0.39 to 0.47. It is clearly observed that, with N splits, the value of FDR is 99.99% with very low FAR of 0.001%. Similar trends have been observed for Y family as shown in Fig. 4(b). As Z dataset is highly normalized. Here, the FDR value of random samples varies from 90% to 94% with a quite low value of FAR i. e., 0.18% only. The trend is observed, as the data unbalancing problem is less in this dataset as compared to the Back blaze dataset. Here, good samples are three times more than bad samples and for N splits; it shows the best results with 99.99% FDR and 0.001% FAR.

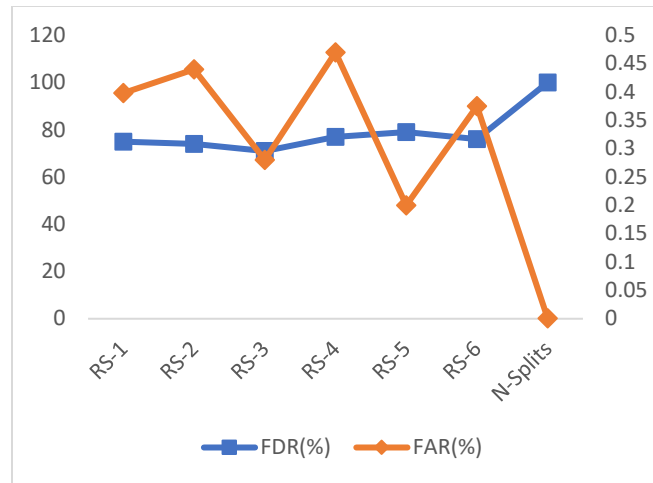


Figure 4(a): FDR and FAR of different random combinations of the training set for the X family

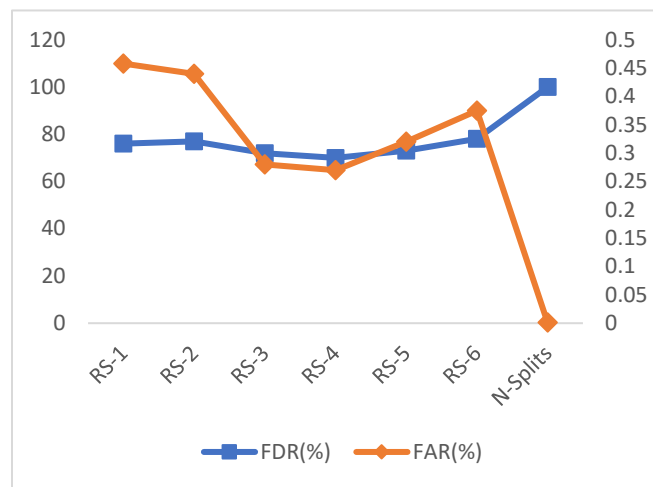


Figure 4(b): FDR and FAR of different random combinations of the training set for the Y family

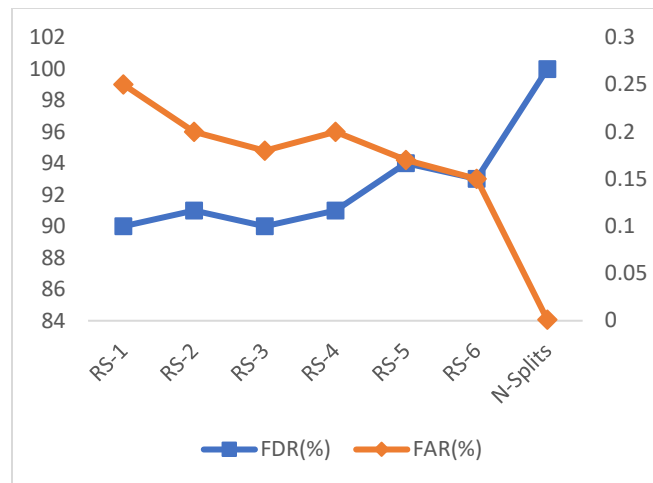


Figure 4(c): FDR and FAR of different random combinations of the training set for the Z family

As it has been explained earlier that the least important features are extracted using RFE method, as a pre-processing step before testing our model for failure predictions. Figs. 5(a), 5(b), 5(c) shows the impact of SMART features' importance on failure prediction for family 'X', 'Y' and 'Z'. It shows the near to linear relationship between extracted features and accuracy of our model. As unimportant features are eliminated, FDR increases from 86.5% to 96% with the low FAR of 0.1% for X family. Similar trends have been observed for 'Y' and 'Z' family.

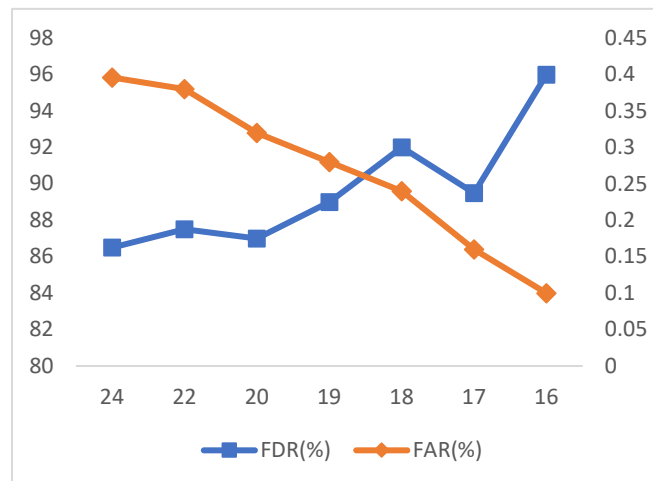


Figure 5(a): FDR and FAR of different features combinations excluded using RF-RFE for the X family

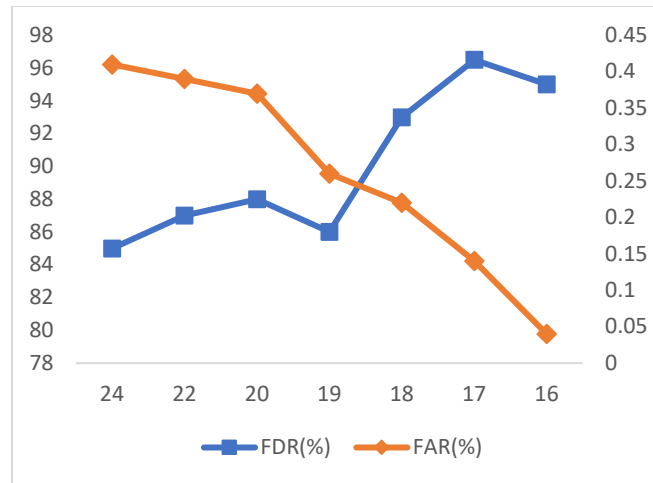


Figure 5(b): FDR and FAR of different features combinations excluded using RF-RFE for the Y family

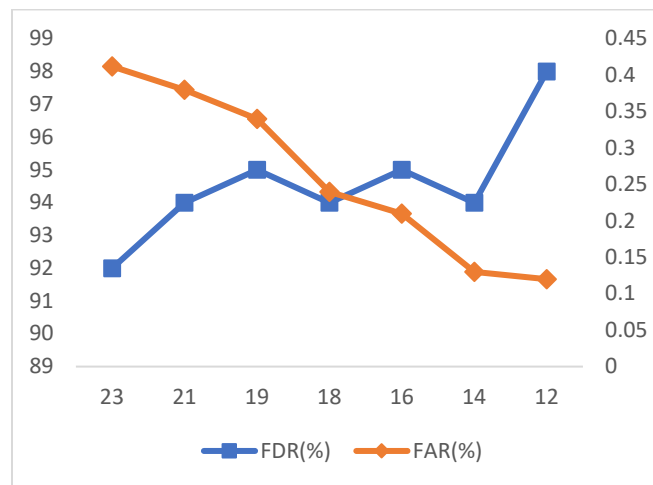


Figure 5(c): FDR and FAR of different features combinations excluded using RF-RFE for the Z family

As a constant change of SMART parameters is helpful for the health status prediction of hard drives. Fig. 6, provides the details of FDR and FAR for multiple arrangements of basic features along with their change rates. Fig. 6(a) illustrates the detection rate and false alarms generated by our model with basic features along with a 1-day, 2-day change rates for X family. Here FDR changes slowly, performs great. FAR follows similar trends as well. So, it is clear that FDR is good with basic features as 99.0% but FAR is high. But with 7-day change rate, FAR is low as 0.09% with FDR of 93%. The same trend follows with “Y” family. Therefore, we have included a 7-day change rate as a parameter in the dataset.

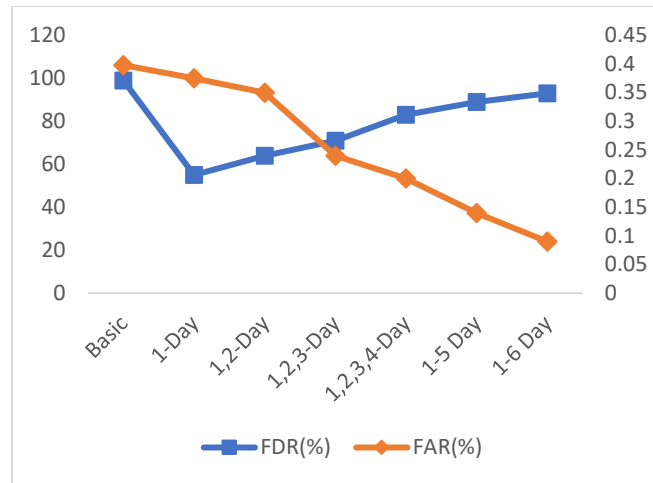


Figure 6(a): FDR and FAR of different arrangements of basic features along with their change rates for the X family

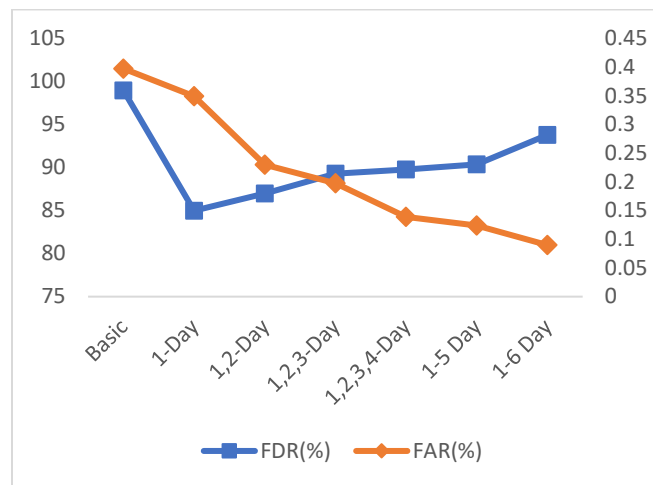


Figure 6(b): FDR and FAR of different arrangements of basic features along with their change rates for the Y family

Fig. 6(c) illustrates the detection rate and false alarms generated by our model with basic features along with 3-hour, 6-hour, 12-, 24-, 48 and 56-hours change rates for Z family. Basic features along with 56-hour change-rate have accomplished with a highest detection rate of 99%, but with high false alarms. 24-h and 48-h change rates are giving a good detection rate with low false alarms. Therefore, we have included a 24-hour change rate as a parameter in the dataset.

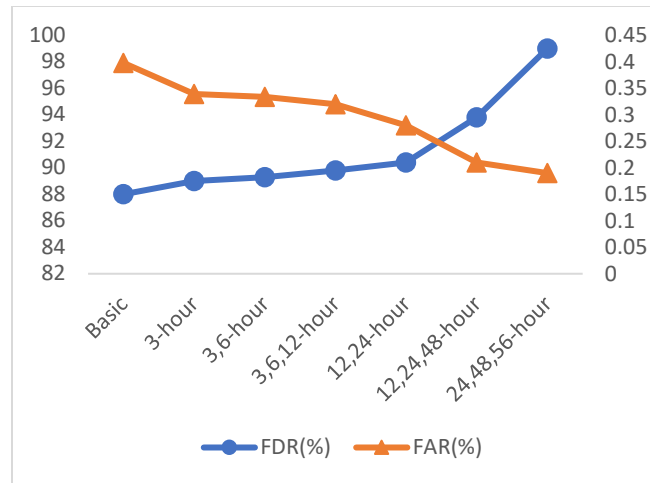


Figure 6(c): FDR and FAR of different arrangements of basic features along with their change rates for the Z family

Fig. 7, shows the ROC curve plotted between FDR vs FAR for performance optimization. It can be clearly visualized that when the number of voters has increased, FDR increases while FAR decreases monotonically. For votes less than $n/2$, FDR is quite low and FAR is high. When the number of voters has increased to $n/2$ or more than $n/2$, FDR increases with the reduction of FAR gradually. For X family, when voters are less than 51, there is more change in the value of FDR and FAR. Afterward, FDR increased much more than FAR, false alarm value reduces from 0.12 to 0.90 with FDR value of 93% to 97.5%. A similar trend has been observed in family Y and Z. FAR decreases with more leap than FDR, with an increasing number of voters. Best results when a number of voters are 51 with FDR=99.99% and FAR 0.01%. Due to lack of space, explanation of results for Y family is not elaborated.

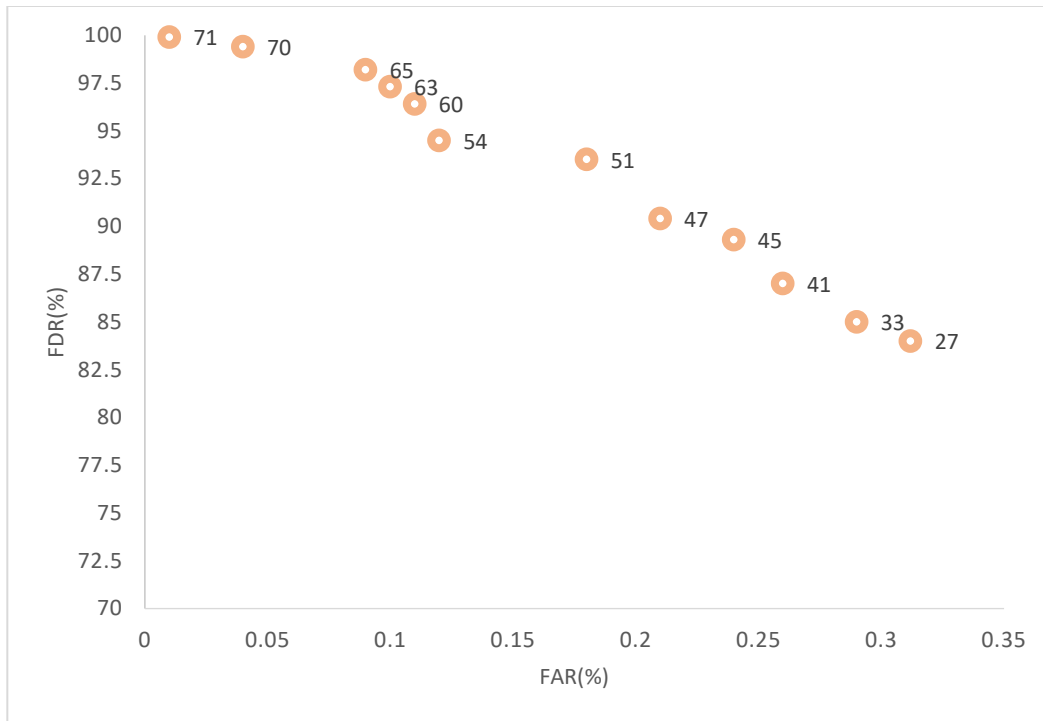


Figure 7(a): Comparison of ROC curve for different Voters for the X family

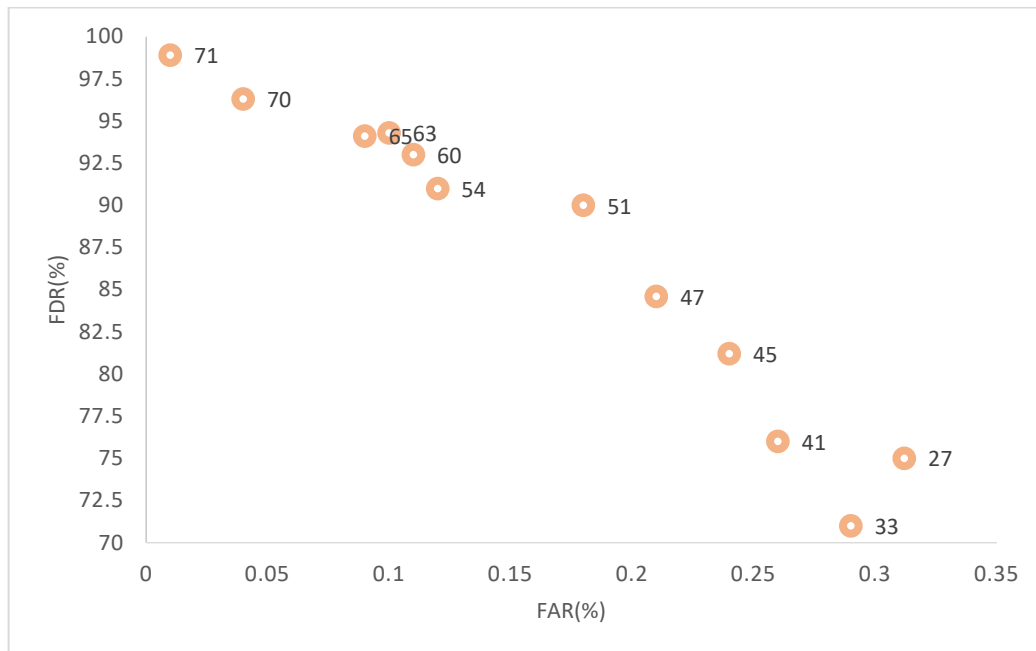


Figure 7(b): Comparison of ROC curve for different Voters for the Y family

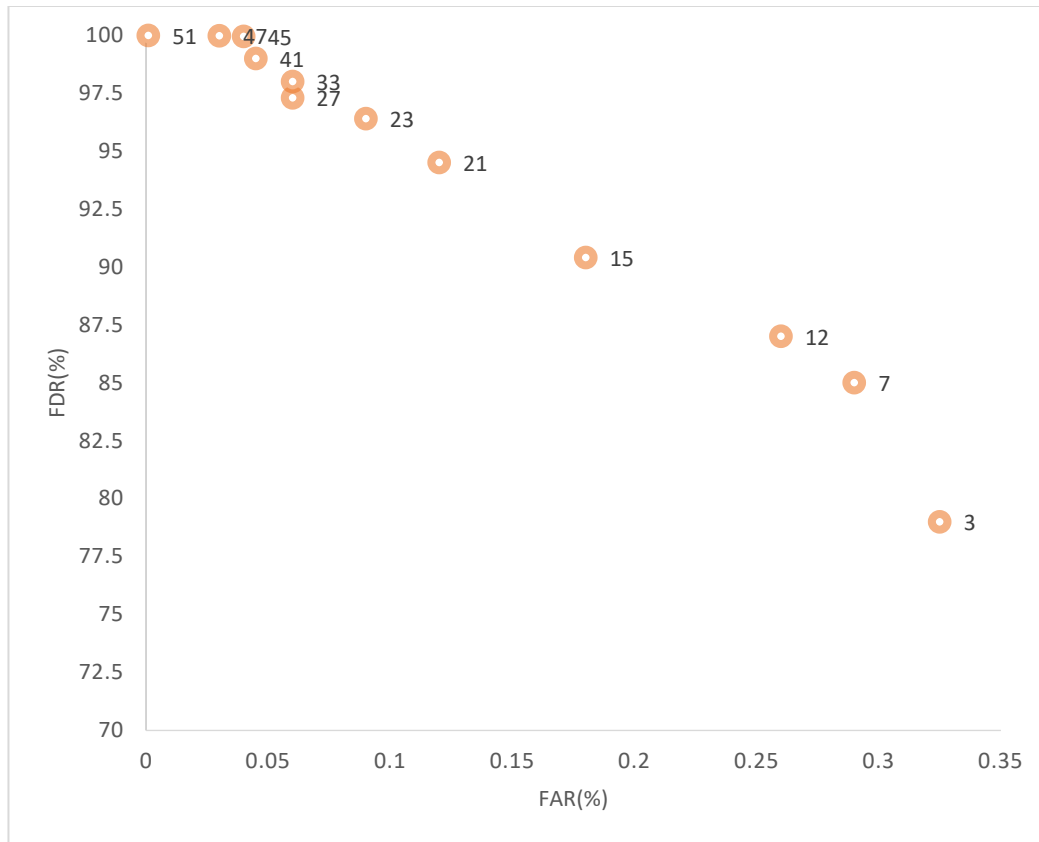


Figure 7(c): Comparison of ROC curve for different Voters for the Z family

Fig. 8 shows that the impact of time window on the failure detection rate. We have tested 7-days, 30-days, 240-days and 365-days time window for 'X' and 'Y' family and 12-hours, 24-h, 96-h, 240-h and 312-h time window for 'Z' family. The ROC- curve shows that results are better for a large time window. As time window increases, a number of samples for training the model increases. Thus, our model learns from every possible sample of a hard disk drive. For the short window interval, fewer samples are provided for training, generating more false alarms. As it can be observed that for 7-days' time window, FDR is 90% and FAR is 0.35%. But with 15-days time window, FDR increases to 95%, but FAR remains almost the same. Results are constant when time window of 120-days or more is taken for X family. However, Z family shows quite different results, with just 24-hour time window its shows best results with 97.5% FDR and 0.1% FAR. By increasing the time window in the Z family, FDR is increasing, but FAR is increasing as well.

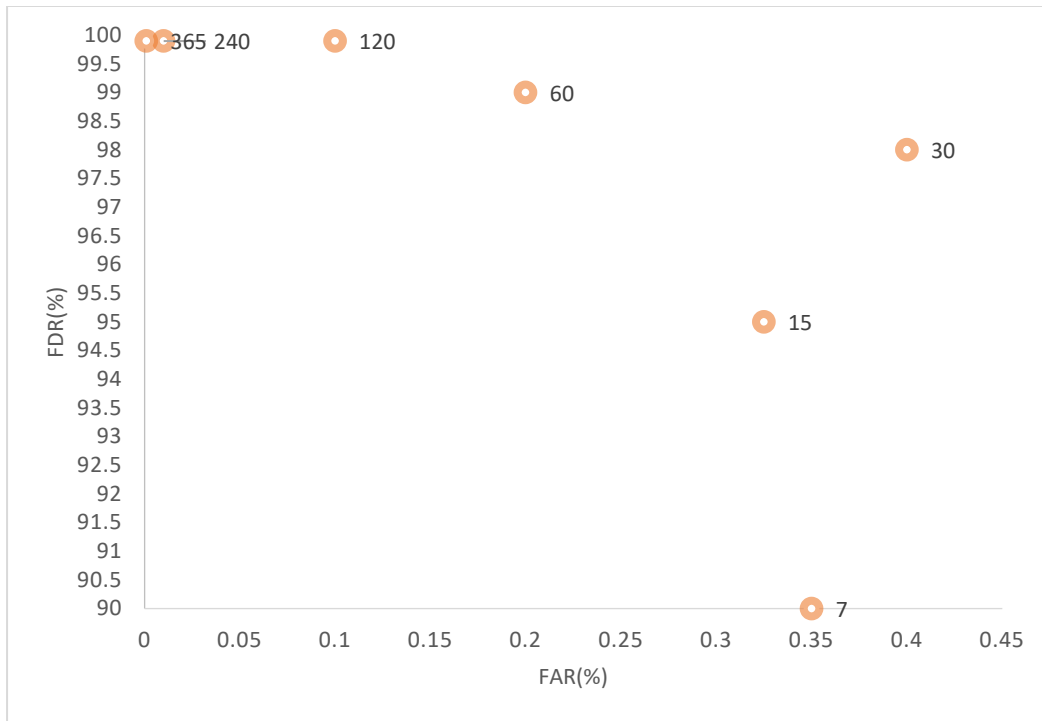


Figure 8(a): Comparison of ROC curve for different Time Window for the X family

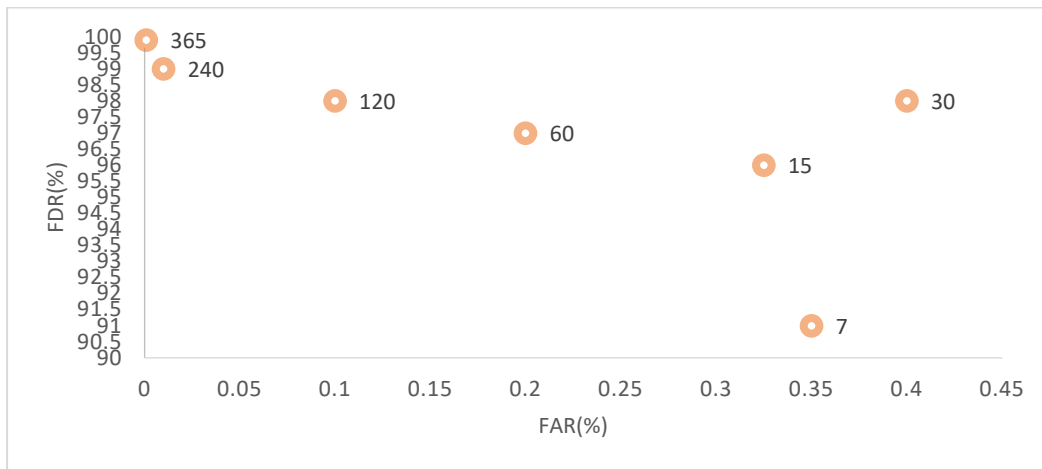


Figure 8(b): Comparison of ROC curve for different Time Window for the Y family

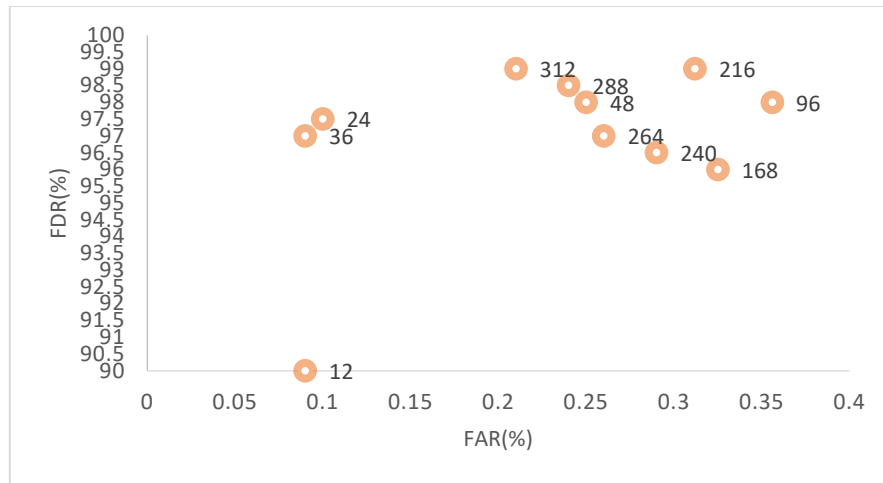


Figure 8(c): Comparison of ROC curve for different Time Window for the Z family

Fig. 9 shows the lead time of HDDs when it is predicted to fail using our model. As it can be clearly observed that more than 900 HDD are predicted at lead 10 days before failure or 30 days before failure. This time allows the backup or data migration process to be triggered in a timely manner.

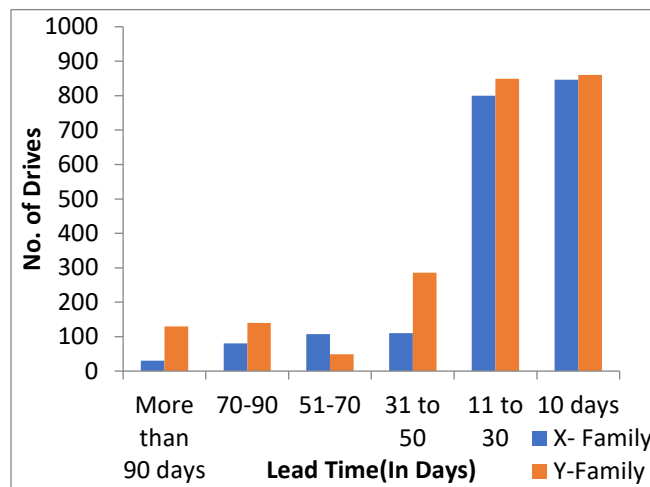


Figure 9(a): Lead Time Prediction on X and Y family

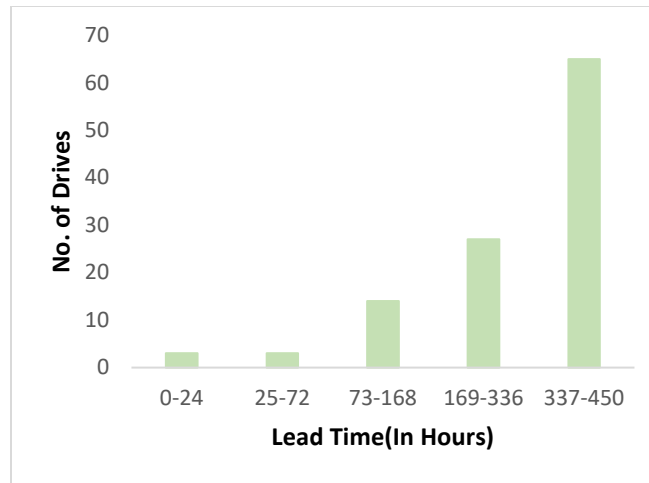


Figure 9(b): Lead Time Prediction on Z family

6.3 Comparison with state of the art

This section compares the prediction performance for HDDs by using our method with the present state of the art methods (details of these methods in Section 2). Due to space limitation, we have compared our method with basic random forest, part voting based random forest and recurrent neural network.

In Fig. 10 part-voting RF and RNN results are compared by using ROC curve obtained for all family of X, Y, and Z dataset. On family X the basic Random Forest is giving low FAR but corresponding FDR is also low as shown in Fig. 10(a). However, we have achieved FDR up to 99.9% with a minimal FAR value of 0.01% only with the application of our model (voting-based decision tree). Similarly, RF-part voting method has not performed up to the mark. FDR value ranges from 72-74%, which is quite low with a high FAR rate (due to data unbalancing problem). Thus, we can say that part-voting RF outperforms basic RF but it is quite lower than our approach. A similar result has been obtained for the 'Y' family shown in Fig. 10(b).

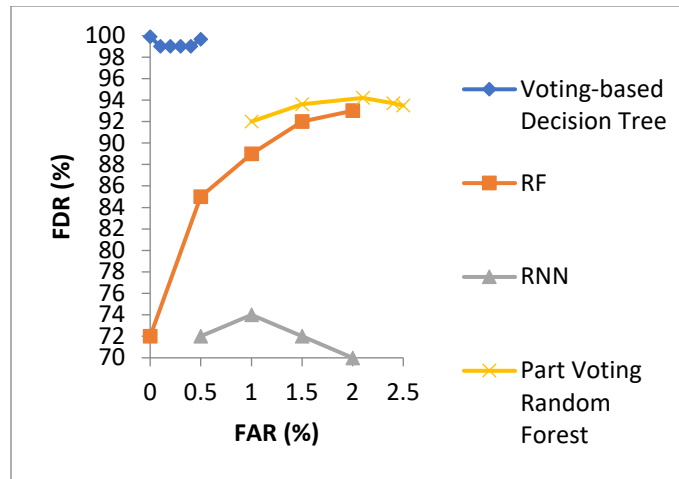


Figure 10(a): Comparison of ROC-Curve for the X family

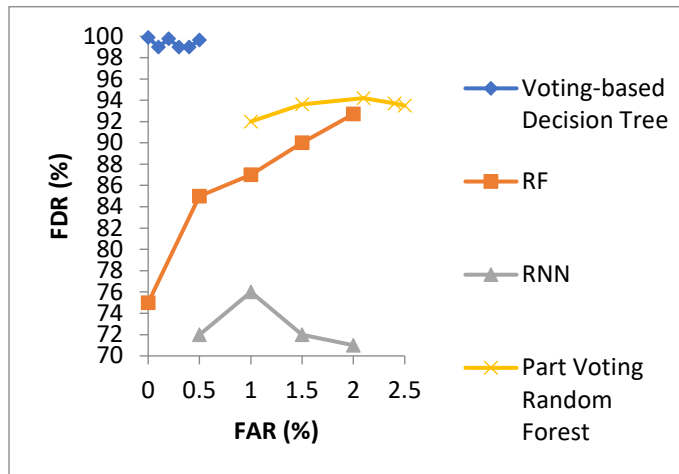


Figure 10(b): Comparison of ROC-Curve for the Y family

Fig. 10(c) compares the results of the balance data set and normalized features. So, for this dataset, almost every other method performs well, having high-value FDR but FAR is also high. In that case, our method outperforms all the previous literature.

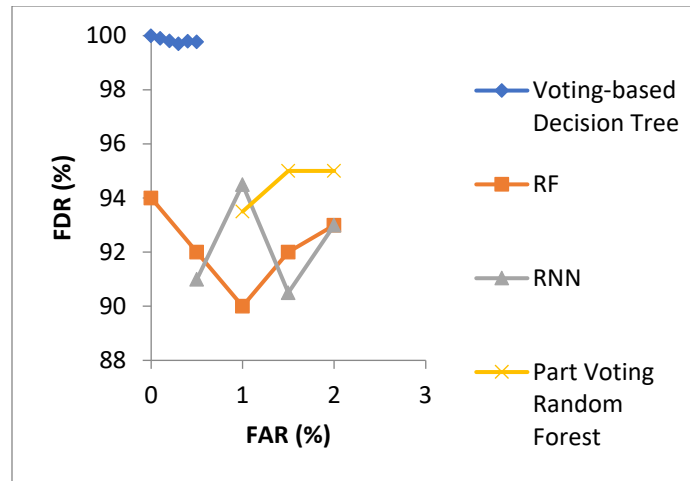


Figure 10(c): Comparison of ROC-Curve for the Z family

Fig. 11 shows the Acc_{hl} and Tol_{hlt}^{acc} evaluation for X family. We have selected random samples of failed or predicted to be failed drives and provide inputs to R-CNN. Health degree is predicted into 6 classes (the reason behind selected 6 classes can be seen in Section 3.6). Fig. 11(a) shows the results of health degree prediction for R-CNN. As maximum HDDs are predicted with exact health degree, the results are clustered around ground realities. Fig. 11(b) and Fig. 11(c) shows that RNN and Multi-NN are not able to provide reasonable results in health degree prediction. The results are much scattered with respect to ground reality, tolerance accuracy Tol_{hlt}^{acc} is low here, as the some of the HDDs are predicted in level 2 instead of level 1. But it is still acceptable.

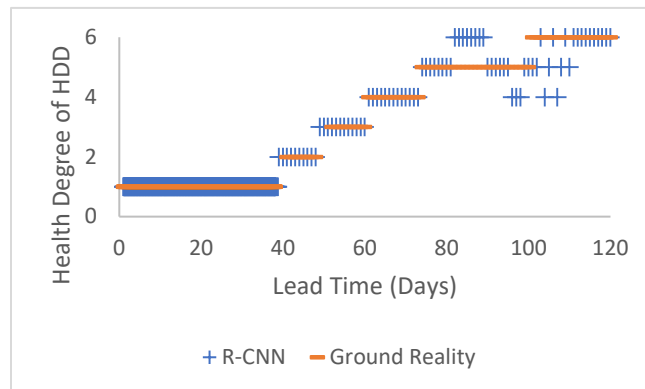


Figure 11(a): Health Degree assessment by using CNN on the X family

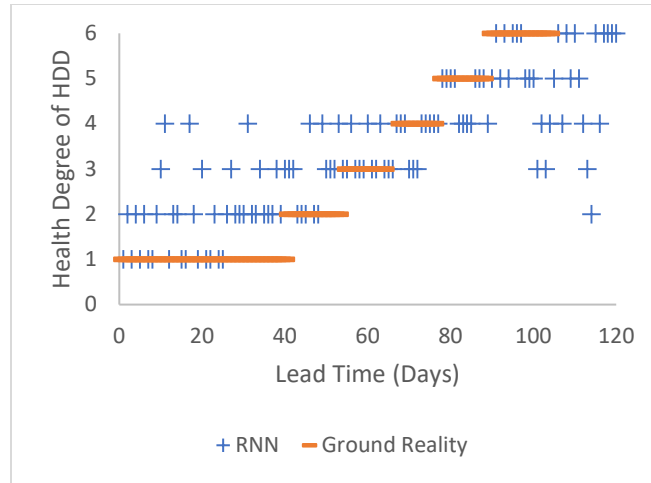


Figure 11(b): Health Degree assessment by using RNN on the X family

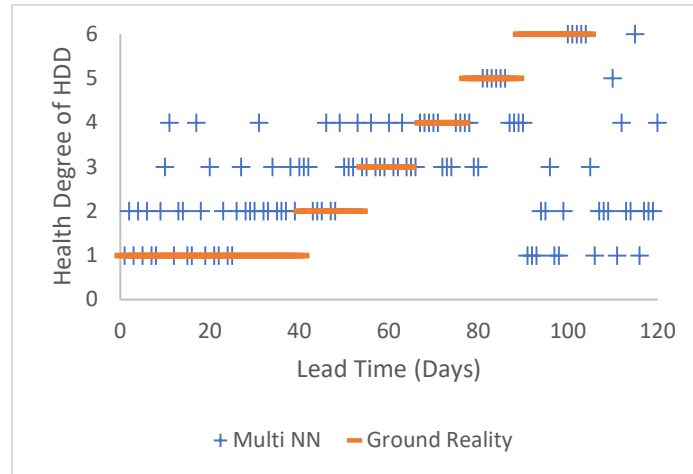


Figure 11(c): Health Degree assessment by using Multi-NN on the X family

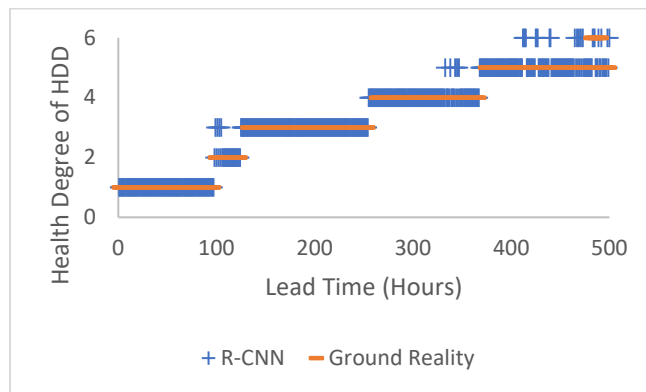


Figure 12(a): Health Degree assessment by using CNN on the Z family

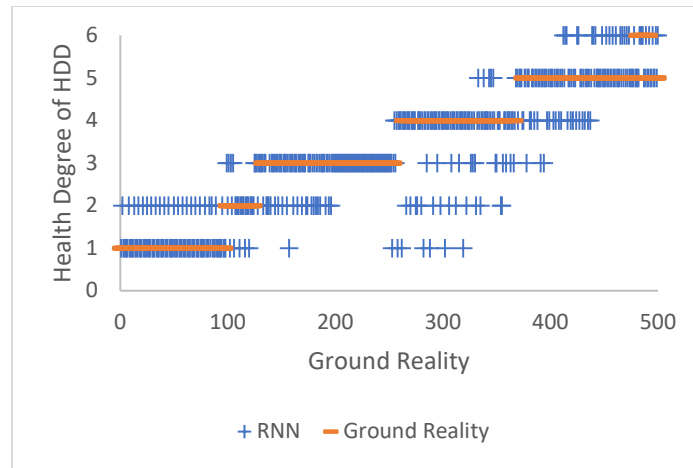


Figure 12(b): Health Degree assessment by using RNN on the Z family

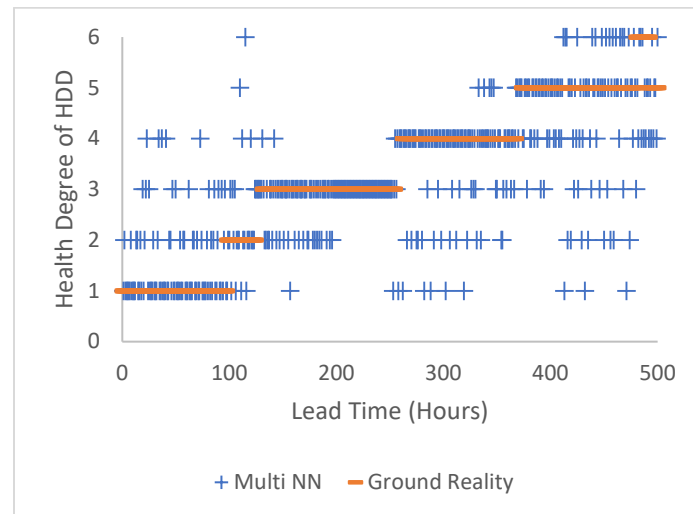


Figure 12(c): Health Degree assessment by using Multi-NN on the Z family

Similar results for the Z family has been observed in Figs. 12(a), 12(b), 12(c). R-CNN provides the maximum accuracy for the prediction of the health status of hard drives in comparison to the RNN and multi-NN.

Fig. 13. Comparison of execution time for prediction of HDD failure of the Voting based decision tree, RNN, Random forest and part-voting random forest for family “X”, family “Y” and family “Z”. This shows the time efficacy of our model; it outperformed the state-of-the-art methods in terms of time complexity also.

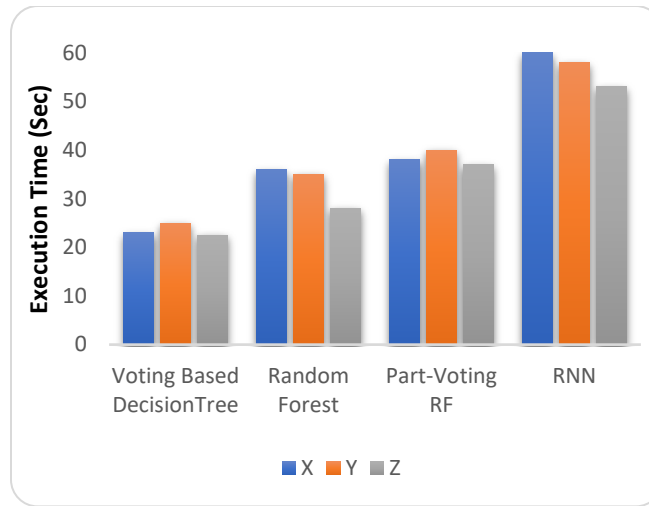


Figure 13: Comparison of execution time for prediction of HDD failure of Voting based decision tree, RNN, Random forest and part-voting random forest for family “X”, family “Y” and family “Z”

7 Conclusion

This paper described a novel homogeneous method for failure prediction, Lead time estimation and Health degree examination of Hard disks. We have utilized the combination of Machine learning and deep learning approach for the prescribed tasks. Moreover, a unique algorithm for data unbalancing problem has been proposed. The model is trained on two publicly available datasets. The methodology is based on voting-based decision trees, BSA algorithm, and Recurrent neural network. We have tested our method by using a dataset of real-life. The exhaustive experimental results are provided which shows that our method outperforms the state-of-the-art techniques. We have achieved an FDR value of 99.99% with a quite low FAR value of 0.01%. After the failure prediction, we have estimated its lead time. Afterward, the power of deep learning is exploited and the health status of soon to fail drives is examined, in 6 classes, ranging from safe to in- danger drives. During the testing, we have observed that for the yearly dataset at least 7 days data is required for achieving the best results. Our future work will be focused on the development of a method that uses robustness of pure, dense deep learning to minimize the time window problem, stated above.

References

- Bairavasundaram, L. N.; Arpacı-Dusseau, A. C.; Arpacı-Dusseau, R. H.; Goodson, G. R.; Schroeder, B.** (2008): An analysis of data corruption in the storage stack. *ACM Transactions on Storage*, vol. 4, no. 3, pp. 1-28.
- Bairavasundaram, L. N.; Goodson, G. R.; Pasupathy, S.; Schindler, J.** (2007): An analysis of latent sector errors in disk drives. *ACM SIGMETRICS Performance Evaluation Review*, vol. 35, no. 1, pp. 289.

Beach, B. (2014): Hard drive smart stats. <https://www.backblaze.com/blog/hard-drive-smart-stats/>.

Eckart, B.; Chen, X.; He, X.; Scott, S. L. (2008): Failure prediction models for proactive fault tolerance within storage systems. *IEEE International Symposium on Modeling, Analysis and Simulation of Computers and Telecommunication Systems*, pp. 1-8.

Gastwirth, J. L. (1972): The estimation of the Lorenz curve and Gini index. *The Review of Economics and Statistics*, pp. 306-316.

Granitto, P. M.; Furlanello, C.; Biasioli, F.; Gasperi, F. (2006): Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products. *Chemometrics and Intelligent Laboratory Systems*, vol. 83, no. 2, pp. 83-90.

Hamerly, G.; Elkan, C. (2001): Bayesian approaches to failure prediction for disk drives. *Proceedings of the Eighteenth International Conference on Machine Learning*, pp. 202-209.

Hamerly, G.; Elkan, C.; Diego, S.; Jolla, L. (2001): Bayesian approaches to failure prediction for disk drives. *Proceeding of the Eighteenth International Conference on Machine Learning*, pp. 202-209.

Huang, S.; Fu, S.; Zhang, Q.; Shi, W. (2015): Characterizing disk failure with quantified disk degradation signatures: an early experience. *IEEE International Symposium on Workload Characterization*, pp. 150-159.

Hughes, G. F.; Murray, J. F.; Kreutz-Delgado, K.; Elkan, C. (2002): Improved disk-drive failure warnings. *IEEE Transactions on Reliability*, vol. 51, no. 3, pp. 350-357.

Khan, G.; Farooq, M. A.; Hussain, J.; Tariq, Z.; Khan, M. U. G. (2019): Categorization of crowd varieties using deep concurrent convolution neural network. *2nd International Conference on Advancements in Computational Sciences*, pp. 1-6.

Li, J.; Ji, X.; Jia, Y.; Zhu, B.; Wang, G. et al. (2014): Hard drive failure prediction using classification and regression trees. *44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pp. 383-394.

Longadge, R.; Dongre, S.; Malik, L. (2013): Class imbalance problem in data mining: review. *International Journal of Computer Science and Network*, pp. 1305-1312.

Ma, A.; Traylor, R.; Douglass, F.; Chamness, M.; Lu, G. et al. (2015): RAID Shield: characterizing, monitoring, and proactively protecting against disk failure. *ACM Transactions on Storage*, vol. 11, no. 4, pp. 241-256.

Mak, C. W.; Phongtharapat, S.; Suchatpong, C. (2014): U.S. Patent No. 8,698,492. Washington, DC: U.S. Patent and Trademark Office.

Murray, J. F.; Hughes, G. F.; Kreutz-Delgado, K. (2005): Machine learning methods for predicting failure in hard drives: a multiple-instance application. *Journal of Machine Learning Research*, vol. 6, pp. 783-816.

Pecht, M.; Tuchband, B.; Vichare, N.; Ying, Q. J. (2007): Prognostics and health monitoring of electronics. *International Conference on Thermal, Mechanical and Multi-Physics Simulation Experiments in Microelectronics and Micro-Systems*, pp. 1-8.

- Pinheiro, E.; Weber, W. D.; Barroso, L. A.** (2007): Failure trends in a large disk drive population. *Proceedings of the 5th USENIX conference on File and Storage Technologies*, pp. 17-29.
- Pitakrat, T.; van Hoorn, A.; Grunske, L.** (2013): A comparison of machine learning algorithms for proactive hard disk drive failure detection. *Proceedings of the 4th international ACM Sigsoft Symposium on Architecting Critical Systems*, pp. 1-10.
- Queiroz, L. P.; Rodrigues, F. C. M.; Gomes, J. P. P.; Brito, F. T.; Chaves, I. C.** (2016): A fault detection method for hard disk drives based on mixture of Gaussians and nonparametric statistics. *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 542-550.
- Rincón, C. A.; Pâris, J. F.; Vilalta, R.; Cheng, A. M.; Long, D. D.** (2017): Disk failure prediction in heterogeneous environments. *International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, pp. 1-7.
- Salfner, F.; Lenk, M.; Malek, M.** (2010): A survey of online failure prediction methods. *ACM Computing Surveys*, vol. 42, no. 3, pp. 1-42.
- Schroeder, B.; Gibson, G. A.** (2007): Disk failure in the real world: what does an MTTF of 1, 000, 000 hours mean to you? *5th USENIX Conference on File and Storage Technologies*, pp. 1-16.
- Shen, J.; Wan, J.; Lim, S. J.; Yu, L.** (2018): Random-forest-based failure prediction for hard disk drives. *International Journal of Distributed Sensor Networks*, vol. 14, no. 11.
- Strom, B. D.; Lee, S.; Tyndall, G. W.; Khurshudov, A.** (2007): Hard disk drive reliability modeling and failure prediction. *IEEE Transactions on Magnetics*, vol. 43, no. 9, pp. 3676-3684.
- Vishwanath, K. V.; Nagappan, N.** (2010): Characterizing cloud computing hardware reliability. *Proceedings of the 1st ACM symposium on Cloud Computing*, pp. 193-204.
- Wang, Y.; Ma, E. W.; Chow, T. W.; Tsui, K. L.** (2013): A two-step parametric method for failure prediction in hard disk drives. *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 419-430.
- Wang, Y.; Miao, Q.; Ma, E. W.; Tsui, K. L.; Pecht, M. G.** (2013): Online anomaly detection for hard disk drives based on mahalanobis distance. *IEEE Transactions on Reliability*, vol. 62, no. 1, pp. 136-145.
- Wang, Y.; Miao, Q.; Pecht, M.** (2011): Health monitoring of hard disk drive based on Mahalanobis distance. *Prognostics and System Health Management Conference*, pp. 1-8.
- Xin, Q.; Miller, E. L.; Schwarz, T.; Long, D. D. E.; Brandt, S. A. et al.** (2003): Reliability mechanisms for very large storage systems. *20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologie*, pp. 146-156.
- Xu, C.; Wang, G.; Liu, X.; Guo, D.; Liu, T. Y.** (2016): Health status assessment and failure prediction for hard drives with recurrent neural networks. *IEEE Transactions on Computers*, vol. 65, no. 11, pp. 3502-3508.
- Yang, W.; Hu, D.; Liu, Y.; Wang, S.; Jiang, T.** (2015): Hard drive failure prediction using big data. *IEEE 34th Symposium on Reliable Distributed Systems Workshop*, pp. 13-18.

Yen, S. J.; Lee, Y. S. (2009): Cluster-based under-sampling approaches for imbalanced data distributions. *Expert Systems with Applications*, vol. 36, no. 3, pp. 5718-5727.

Zhu, B.; Wang, G.; Liu, X.; Hu, D.; Ma, J. (2013): Proactive drive failure prediction for large scale storage systems. *IEEE 29th Symposium on Mass Storage Systems and Technologies*, pp. 1-5.

Zimmer, V.; Rothman, M. (2007): U.S. Patent Application No. 11/242.