National College of Ireland

A Stable Model to Predict the Hard disk failure

MSc Research Project
Data Analytics

Venkata Krishnan Mittinamalli Thandapani
x16134311

School of Computing
National College of Ireland

Supervisor: Dr. Urvesh Bhowan

National College of Ireland
Project Submission Sheet – 2017/2018
School of Computing

Student Name:	Venkata Krishnan Mittinamalli Thandapani
Student ID:	x16134311
Programme:	Data Analytics
Year:	2016
Module:	MSc Research Project
Lecturer:	Dr. Urvesh Bhowan
Submission Due Date:	11/12/2017
Project Title:	A Stable Model to Predict the Hard disk failure
Word Count:	7523

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature:	
Date:	11th December 2017

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
3. Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Stable Model to Predict the Hard disk failure

Venkata Krishnan Mittinamalli Thandapani
x16134311
MSc Research Project in Data Analytics

11th December 2017

Abstract

The increase in the production of digital information every second has raised the demand for storage as its counterpart. Hard disk serves an important storage device but subjected to failure which leads to the loss of data. To improve its reliability several researches were put forward, which didn't provide a satisfactory result and had no real-time implication of their system. The purpose of this research is to propose Disk Failure Prediction (DFP) models based on Random Forest (RF), Feed Forward Neural Network (FFNN) and unsupervised K-means clustering, with a real-time Self-Monitoring System (SMS) built on a predictive model showing the most stable and reliable performance. The performance of the model were evaluation under various test cases such as providing samples of different sizes, taking timing into account and considering voting in the predicted result. The RF model performed best under this test cases with a maximum DFP rate of 99% with 0.01% False Alarm Rate(FAR) which is superior to state of the art model such as Decision Tree (DT).

Keywords: Random Forest, feedforward neural network, decision tree, K-means clustering, smart monitoring system, h2o, hard disk failure.

Contents

1 Introduction

With the blooming digital world, creating large amounts of digital information has increased the demand for storage. It has led to a 7% increase in storage market every year (Kaplan; 2008). This demand has caused a massive shipping of hard disk, nearly 104 million hard disks were shipped in the third quarter of 2017 (Peng; 2017). This hard disk is highly reliable, but its failure may lead to a drastic loss of useful information of users. And a heavy economic loss for the service provider is observed during the downtime. It is estimated (Gehrer; 2016) that if a data center is missing in action it would cost $9,000 US every minute. This Significance of the hard disk demands it to be more reliable in functionality. But it is observed that the disk remains as the most often failing component causing that downtime in a data center which drives the research interest on it (Botezatu et al.; 2016).

To improve the reliability of the disk, the pioneer of disk manufacturer IBM, introduced Self-Monitoring Analysis and Reporting Technology systems (SMART) (Allen; 2004). This SMART system is designed to predict the disk failure in advance so, as to facilitate the user to save their data before the actual failure could occur. This system works based on threshold method, the features of the disk where continuously monitored by the SMART and its compared with the threshold level provided by the disk manufacturer. An increase in the feature value above the threshold would indicate the user about the upcoming failure. The SMART threshold value varies among the manufacturers. According to Allen (2004), this method provided a very low prediction of 3%- 10% with a FAR of 0.01%. There is not even 50% guaranteed chance of correctly predicting the *will fail disk*. More importance is given to the FAR to be maintained low as an increase in this value could have a negative impact on the manufacturer. As they need to replace

the disk which fails before the warranty.

The SMART system provided a very low failure prediction, a great deal of researchers had made the contribution in improving the prediction of the disk failure with reduced FAR. This was done by building model based on SMART attributes using some of the statistical and machine learning methods (Murray et al.; 2005, 2003; Wang et al.; 2011; Queiroz et al.; 2016; Li et al.; 2017).

These built models had many drawbacks, most of this model showed a low prediction with high FAR than the SMART system. Nearly half of them used a very small data set which defers from the real-world hard disk data. The predicted results were not stable with the varying data sizes. They do not depict any practical application of the model built for prediction. The existing system has several features to be improved upon.

The objective of this research is to deploy machine learning based models that will perform better than the existing methods in terms of higher FPR with lower FAR, maintaining its stability under various circumstances. A Classification and clustering based models such as ensembled RF, unsupervised K-means clustering and supervised FFNN will be implemented to get improvement in performance in terms of prediction and stability. They will be tested on a dataset coming from real-world data center used by the previous researcher Li et al. (2017) to have a comparative study on their performance. A SMS system will be built for a real-time monitoring of system disk status using the best performing machine learning algorithm.

The remainder of this paper is organized as follows. The previous researchers work on DFP are briefly reviewed in Section II. The dataset description and summary of our proposed methods with initial adjustment are provided in section III. Implementation of the proposed methods is given in section IV followed by the evaluation of the algorithms in section V. The section VI concludes the paper.

2 Related Work

2.1 The existing smart technology and its cons

The SMART introduced by IBM has a goal to provide valuable time to the user to safeguard their data before the failure of the disk could occur Allen (2004). This system doesn't provide to be a good model as it showed a very low Failure Detection Rate (FDR) with moderate FAR.

Many works were put forward by researchers to improve upon this feeble section of the SMART system with the main intention of achieving higher FDR at a minimal FAR. The work by the researchers in detecting the hard disk failure is divided into three main categories as said my Chaves et al. (2016). The first set of researchers used the hard disk generated log files, but it lacked in providing incisive information. The second set used externally built monitoring system designed using sensors, but the cost of implementation of this system was high. The third set had an inbuilt technology to monitor the status of the disk with minimal prediction rate of 3% with a FAR of 0.01% named as the SMART system (Allen; 2004). Most of the researchers followed the third set and used the SMART attributes collected by the system to achieve their results.

2.2 Consolidation of statistical and machine learning techniques

The improvement on this SMART system was first conducted by Hughes et al. (2002) using Walcoxon rank-sum test by considering the SMART attributes to be non-parametrically distributed. The method showed a higher FAR of 0.5% showing a detection of 40%-60% which is higher than the default SMART system. This system was implemented using the OR-ed single variant and multivariant rank-sum test. The implemented system was not good for general learning and the dataset used has a very minimal count of data.

Murray et al. (2003) worked on the improvement of this system by including feature selection on data. A comparative study was conducted by him implementing Support Vector Machine (SVM), two nonparametric statistical tests and unsupervised clustering. This implementation showed a major disadvantage with SVM showing deterioration of performance with the selected feature. A high FAR of 1.0% was shown which couldn't be lowered below this range in autoclass. And no further improvement on the 33.2% FDR with 0.5% FAR was achieved by Murray et al. (2003).

With the FAR being very high, reduction of this value was taken into consideration by Murray et al. (2005) as his further work by using non-parametric methods, assuming that the attributes are non-parametrically distributed. A different dataset having 191 failed and 178 good disks making a total of 369 disks were used. To get a better prediction and to have minimum FAR separate groups of algorithms were considered to focus on each of this causes separately. Even the newer data set with different feature selection performed using reverse argument and z-score the prediction of 33.2% with 0.5% FAR was given by the rank-sum method and found to be better than Nave Bayesian classifier(mi-NB) which was newly implemented with a hope to achieve a lower FAR. The attribute consideration still had an effect on SVM which showed 0% FAR at 50% FDR with full attribute and a deprivation with limited attributes. It also took longer time to set SVM than other algorithms. With the SVM performance, it is clear that attribute selection has a major effect on the performance of the model as said my Saeys et al. (2007).

Disagreeing with above researchers (Murray et al.; 2005, 2003; Hughes et al.; 2002), Li et al. (2014) pointed that the good and failed disk considered in Murray et al. (2005) research had different point of origin and data set was smaller and had almost similar count of good and failed disk which won't be the case with real data from a hard disk system. They also showed a very low prediction with greater FAR.

The Controversial argument was raised by Zhao et al. (2010) that relationship between the attributes is an important factor which changes over time and previous researchers (Murray et al.; 2005, 2003; Hamerly et al.; 2001) didn't consider this relationship into account. Zhao et al. (2010) predicted the disk failure using HiddenMarkov-Models (HMMs) and Hidden Semi-MarkovModels (HSMMs) models by considering the relationship between the attribute. A set or individual attributes were considered as a time series observation and log likelihood calculated on the train and test set was done to compute a threshold to predict the disk status. With this consideration, the HMM and HMSS showed a performance of 46% and 30% for single and 0% FAR with 52% prediction for multiple attributes. The performance was superior to (Murray et al.; 2005, 2003).

2.3 Considering anomaly behaviour of hard disk for fault detection

The abnormal behavior is named as an anomaly. Its mostly used in cases where the samples are imbalanced such as the case in rare disease prediction. The methodology to detect the fault in disk using anomaly was first carried out by Wang et al. (2011). He used Mahalanobis distance to determine the abnormality in most of his experiments, the measured mark of the degree of degradation is compared with a threshold to determine the disk status. He used feature selection by conducting a detailed study on the structure and content of the hard disk. The attributes selected by Wang et al. (2011) was different from that of Murray et al. (2003). The anomaly method had a greater prediction accuracy with the selected feature showing a detection capability of 63% with 0.56% FAR which was better than Hamerly et al. (2001). To provide a further improvement, to the Mahalanobis distance a Box-Cox transformation was introduced in his subsequent study (Wang et al.; 2012). Along with attribute selection process Wang et al. (2012) using minimum redundancy maximum relevance (mRMR) and failure model, mechanisms, and effects analysis(FMMEA). Improvement was not seen with all this implemented strategy it only increased the computational cost (Wang et al.; 2014).

The use of Mahalanobis distance for fault detection was opposed by Queiroz et al. (2016) raising an opposite viewpoint by stating that use of the distance approach in the case when HDD doesn't meet normality assumption could show a degradation in performance of the model which was not considered by (Wang et al.; 2014). As a solution for this Gaussian system with a recursive feature elimination for attribute consideration is proposed by Queiroz et al. (2016) for detecting the anomaly in disk behavior. The maximum of 80.59% failure detection with minimal FAR was the result of this case.

2.4 Implemented techniques of machine learning to predict the disk failure

Chaves et al. (2016) opposed the Gaussian method proposed by Queiroz et al. (2016), by stating the difference between failure prediction and fault detection. The numerical quantity defining the rest of the useful life of the disk is given by failure prediction and the finding of this failure in advance is given by false detection which focuses on the anomaly of the disk. The previous researchers Queiroz et al. (2016) and Wang et al. (2014) didn't consider the prediction of failure and gave more importance in fault detection. Queiroz et al. (2016) differentiated his work by building a model using Bayesian Network and named Bayesian Network based Hard Disk Drive (HDD) Failure Prediction(BaNHFaP) which used the Power-On attribute of the disk discriminating itself from those models which used only the remaining useful life. The dataset used had four percentage of failed disk and was provided by Backblaze company.

The above researchers showed a better prediction than the SMART system but most of them were subjected to controversy some of the models by Chaves et al. (2016); Murray et al. (2005, 2003) were black box approach and had inconsistent showing a low prediction with higher FAR. The later researcher Li et al. (2014) mentioned that the SMART attributes changed with the deterioration of the disk which was not considered into account. And had no description of the way the model could be used in the real world. The

data set used are mostly smaller and had a minimum failed disk samples

Considering all this into account Li et al. (2017) used DT for hard disk prediction problem and Gradient boosting regression (GBRT) for monitoring the health degree and got a maximum prediction of 93% at FAR of 0.01% with DT. They considered 13 best features from a real-world dataset after going through three non-parametric statistical tests methods used by Hughes et al. (2002). Though the performance was higher than the rest of the researchers the DT lacked its stability with smaller samples and the prediction was lower than expected with a higher FAR rate of 0.01%. And there were degradation in performance of both DT and GBRT when the hybrid dataset was considered.

2.5 Summary of recent research works on hard disk failure prediction

S.No.	Reseachers	Techniques	Cons with the implemented system	Reference
1	Zheng,Zhao and Lui(2010)	HSMM-and HMM-based approaches	The maximum FDR was 50% which is minimal and needs improvement.	(Zhao et al.; 2010)
2	Miao, Wang and Michael peche (2011)	distance using Mahalanobis	It works with normality assumption and no consideration was made on HDD which doesn't meet this assumption.	(Wang et al.; 2011)
3	Miao,Wang and Michael peche (2012)	A fusion approach on Box-Cox Transformation Mahalanobis	It works with similar assumption of normality and would show degredation in performance with HDD not meeting this assumption.	(Wang et al.; 2012)
4	T. Brito, C. Brito, Queiroz, Radrigues Gomes and Machado (2016)	Mixture of Gaussians	Their consideration was on fault detection and no attempt was made on failure prediction	(Queiroz et al.; 2016)
5	Xinpu, Jing, Zhu, Yuhan and Wang (2014)	Regression and Classification tree	Their performance was limited to FDR of 95% with higher FAR of 0.1%.	(Li et al.; 2014)
6	Wang, Gang, Li, Jing, Rebecca and Li, Zhongwei, Ming (2014)	HDD failure prediction using DT and GBRT	The performance was not stable and the maximum prediction was only 93% with 0.01% FAR	(Li et al.; 2017)

2.6 Tracking down to the implemented system

The disk failure forecast is a classification problem as stated by Li et al. (2014). RF introduced by Breiman (2001) provides both classification and regression which is highly accurate and more robust to noise which provides the estimate of variable importance (Díaz-Uriarte and De Andres; 2006). **The voted result of internally built trees by RF makes the result more precise than other algorithms. It is stated by**

(Li et al.; 2014) that RF based classification could be a better feature work on DFP. On the other hand, the DT used in previous research is subjected to error due to bias and variance which is internally looked after by RF, it reduces variance by taking subgroup of features and diverse samples of data for training. They can maintain bias error with no increase in overfitting.

The RF is used by many researchers to achieve better prediction than state-of-the-arts model such as SVM which is clear from Khalilia et al. (2011) research where he used the RF for a highly imbalanced dataset and compared its performance with bagging, boosting and SVM to show RF prediction accuracy. It was also the case with Wei and Chiu (2002) churn prediction, the RF showed superiority than basic Long Short-Term Memory and Gradient Boosting Trees. These features of RF makes it superior to be used in this research.

The neural network (NNET) was introduced in computing by Limpon (1987). They served as a complex system recognizing image and speech. This complex system is been in use for many years and as a classifier by researchers. Kirkos et al. (2007) used an NNET and DT in his classification of fraud document and found the NNET to provide a better prediction. Many of the researchers (Wang et al.; 2011; Queiroz et al.; 2016) treated disk failure detection as anomaly detection where failed and will fail disk which is less in the count is segregated from the good once. K-means is an unsupervised clustering algorithm introduced by Hartigan and Wong (1979) is subjected to anomaly detection. Due to its simplicity and the high-speed performance alone with the capability to handle huge datasets its been used in classification and anomaly detection, Yuan et al. (2017) and Phua et al. (2010) used k means and got a better result.

3 Methodology

3.1 Dataset description

*T*he dataset[1] we have considered for this research is a real-world dataset with 23,395 disk drives from Seagate model (ST31000524NS) which are enterprise-class drives. 22962 drives among the 23,395 are the *good drives* which don't experience any failure during the entire operation while the data was recorded. The rest 433 drives failed at some point while the data was recorded. The SMART attributes of each *good* and the *failed drives* were recorded every hour over a prolonged period of time which makes each disk to have several samples which we named as Individual-Disk-Samples (IDS). All the *bad disk* has IDS of 20 days before the actual failure which is retained in the dataset but most of the drives which didn't last until 20 days of record period had lesser than 480 IDS. All those drives which didn't fail during the record operation were categorized as the *good drives* and the last seven-days IDS of all those *good drives* were retained in the dataset. This entire IDS of data formed the dataset described by the name "W". The statistics of "W" data set is given in Table 1.

The dataset has thirteen features including the serial number which is like disk-ID representing each individual disk by a unique number. The twelve other features of the disk expect serial number is completely normalized to a scale of (-1 to 1). The output label has (-1 and 1) where "-1" denotes *failed disk* and "1" denotes *good disk*. The dataset is unbalanced with a greater number of *good disk* nearly 50 times higher *good disk* than

[1]Dataset is available at: `http://pan.baidu.com/share/link?shareid=189977&uk=4278294944.`

Table 1: Statistic of "W" dataset

Dataset	**Labels**	**Period**	**No. of Hard disk**	**IDS**
"W"	Failed	20 days	433	158,150
	Good	7 days	22962	3,837,568

Table 2: Attributes of "W" dataset

Attribute	
Raw Read Error Rate	High Fly Writes
Spin Up Time	Temperature Celsius
Reallocated Sectors Count	Hardware ECC Recovered
Seek Error Rate	Current Pending Sector Count
Power On Hours	Reallocated Sectors Count
Reported Uncorrectable Errors	Current Pending Sector Count

the *failed disk*. The complete description of the attributes in "W" is given in Table 2

3.2 Preliminary checks and Dataset pre-processing

The data is completely processed in RStudio. The header for the dataset is provided with the description of the dataset which are then included in the dataset. The summary of the dataset are made to have a clear understanding of the features. There where no null values included in the dataset and were completely normalized. The serial number of the disk was not used in build the model. The label column which describes the drive status was converted to factor to acknowledge the model to treat it as the output column. All the other feature were maintained as a numeric quantity. The dataset is divided into test and train, with 70% of the data in train set and rest 30% in the test set. The samples of this data were used to test and train the model built. Further processing of the data is done based on the research conducted on it, this further processing is clearly described in their specific sections.

3.3 Methods description and parameter adjustment

3.3.1 Random Forest based model

The motivation for using RF is due to its peculiar functionality of generating several internal classifiers and providing the voted result of that classifiers, unlike the DT model which used external voting for increasing prediction accuracy (Khalilia et al.; 2011). Along with that, it runs its own internal validation minimizing the (OOB) error which makes it provide a better prediction (Díaz-Uriarte and De Andres; 2006). In addition to that, It avoids overfitting and uses the importance of features to generate better results.

In this research, we took the help of a package randomForest to build RF in R studio. While building this model the important parameter considered are *ntree* and *mtry* variables of the RF which has to be tuned in accordance with the dataset and

the experiment. The *ntree* determines the number of tree to be build from the dataset provided. The *mtry* is the number of variables to be used in each random tree built in the process. The ntree value was set randomly at the first run of the RF model to get a deeper insight on the error estimate. Figure 1 shows the error estimate with *ntree* value of 900. Based on the error value obtained in this first run the *ntree* is fixed to a value of 100.

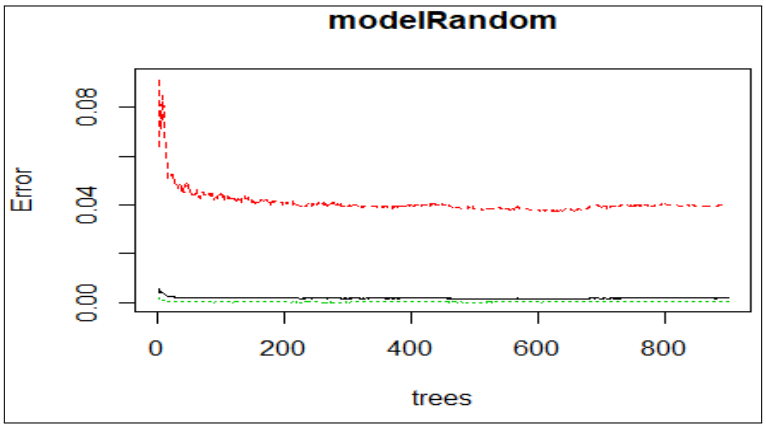

Figure 1: Error estimate for *ntree* value of nine hundred.

The value of *ntree* is minimized as possible as there is no decrease in the error rate after this stage and building further trees did have a major impact on the prediction performance and it increased the computational time to read the additional trees. The value of *mtry* specifies the performance of the model in prediction. To set a proper value of *mtry* the square root of total column in the dataset is considered as a first approach, taken that "X" is the number of column in the dataset.

$$mtry = diskdata(sqrt(ncol(X))) \tag{1}$$

The total of twelve features skipping out the model serial number is selected for building the model. The *mtry* value of 3.4641 is used for the first run. As we proceeded with tuning the RF for a better prediction, using the repeated cross-validation method with grid search which is available in the *caret package* of R. A three folds cross-validation was done twice on the training data to determine the value of new *mtry*. The cross-validation provided the *mtry* value of six which gave the maximum prediction performance with the training data and it was better than the *mtry* set using the formula (1) so rest of the experiment used the *mtry* of six. The values of the rest of all the perimeter in the RF were left to its default value.

3.3.2 K-means clustering model

K-means clustering is an unsupervised learning method. The preference of this method is due to the data set structure which has a minimal number of attributes that are numerical which makes the k-means more suitable unsupervised learning algorithm to be chosen for this problem. And it's simple and suitable for large dataset as it doesn't involve any pre-calculation. Along with this the hard disk failure detection is considered as an anomaly detection by Wang et al. (2012); Queiroz et al. (2016) and work on this

by using different statistical approaches. K-means is a simple and preferred algorithm for detection of the anomaly, which is used to achieve good accuracy in prediction by several researchers (Lima et al.; 2010; Li et al.; 2011). The performance of the unsupervised machine learning in classifying the disk into two categories with the minimal FAR is checked. The k-means clustering is performed in RStudio with its library *k-means*. The important attributes of the k-means clustering are the value of "K" which is the total number of clusters to be formed at the end, the initial number of centroids and the stopping criterion for the k-means clustering.

The value of "K" is assigned as two as we have only two categories as our response variable. The initial number of times the algorithm has to run to get the minimum *within sum of square* is set a random value of five whereas the default value is one. The stopping criterion is given by "iter.max" which is left to its default value of "10". The k-means uses different methods to compute the clusters the default method in which K-means package work is *Hartigan-Wong* which is left unaltered.

3.3.3 Feed-forward neural network based model

The Artificial neural net is a creation with the inspiration of human brain functionality. The self-adaptive feature of NNET according to the input data without explicit interaction. Followed by their flexible nonlinear modeling feature for the real-world complex relationship (Hornik et al.; 1989; Hornik; 1991). Along with their universal functional approximation made their way to be applied in this research. The NNET has shown its precedence when applied to several real-world classifications of fault detection (Bartlett and Uhrig; 1992; Hoskins et al.; 1990) and prediction of bankruptcy (Leshno and Spector; 1996; Tam and Kiang; 1992).

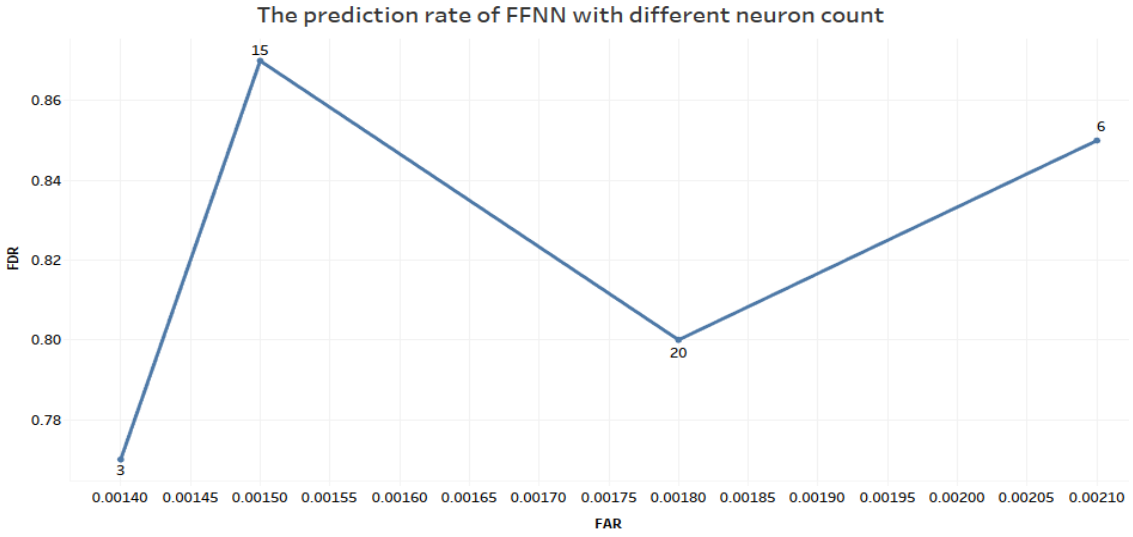

Figure 2: The graphical illustration showing FDR and FAR values for different counts of neurons.

The FFNN was implemented in RStudio using the *nnet* package available in R. This package provides a simple FFNN with a single hidden layer. The number of input neurons is equal to the number of input variables. In this case, the input neurons were

set to twelve and the output layer has a single neuron as there was only a single output variable with two categories. The selection of optimal neurons in the hidden layer is an important task. With many hidden layers, there may be the problem of overfitting and with lesser number of hidden neuron, there may be the problem of underfitting with a poor prediction result. The optimal number of neurons in the hidden layer is selected by trial and error method. For this, a sample of 500,000 rows from "W" dataset is used to train and test the model with varying neuron count as shown in Figure 2.

The optimal number of fifteen neurons is chosen in the hidden layer which showed a good prediction with minimal FAR. With the increase of neurons in the hidden layer beyond the input parameter, the True Positive Rate (TPR) increased with a gradual degradation in FAR. The number of neuron were selected in such a way that the FAR remains minimal with a good prediction rate. The maximum iteration run during the training of the model was fixed to the default value of hundred to minimize the time required to run the training and most of them converged before hundred iterations.

4 Implementation

4.1 Overview

The hard disk prediction has been a research interest for many, the recent research was carried out using DT by (Li et al.; 2017), though the DT showed a better performance than the previous researchers it had degradation in performance with smaller datasets, showing unstable performance with varying sample sizes. We built models using both supervised and unsupervised learning algorithms such as RF, FFNN, and k- means clustering to improve upon the existing prediction capability with minimum false alarm rate and to get a stable performance. The implemented models were checked under different conditions such as samples of data of various sizes, considering the timing into account with the training data, and conducting voting in the predicted results of the algorithm to compute their efficiency under different conditions. The models were implemented in RStudio with all the attributes set to the values defined in section 3.3. An SMS system is implemented for a real-time self-monitoring using the best performing machine learning algorithm.

4.2 The model's stability with varying sample sizes

The stability in the performance of the model could be checked by varying the sample size of the dataset used to train the model. As the sample size contributes to change in the performance of the model in the previous research conducted by Li et al. (2017), where the performance of DT and GBRT degraded with smaller sample size. To check this effect on the algorithm considered, different sample sizes were used to build the model and their performance in prediction was checked. Following the section 3.2, we started with randomly segregating the IDS of data and then sampling the data to different sizes. The minimum size of the sample we used was 1000,00 which was from 417 failed disk and 22594 good disk. The samples were then increased in such a way that the count of disk increased in the samples taken. Eight different samples of a various size were considered. Every sample was then split into train and test set, 70% of the data from the sample was treated as a training set which was used to build the model and the rest 30% of the data was used to test the model. The three different model where built

using the RF, FFNN, and k-means clustering with all the attributes of the algorithm configured from the section 3.3. Figure 3 illustrates a simplified overall view of building a model using samples of data.

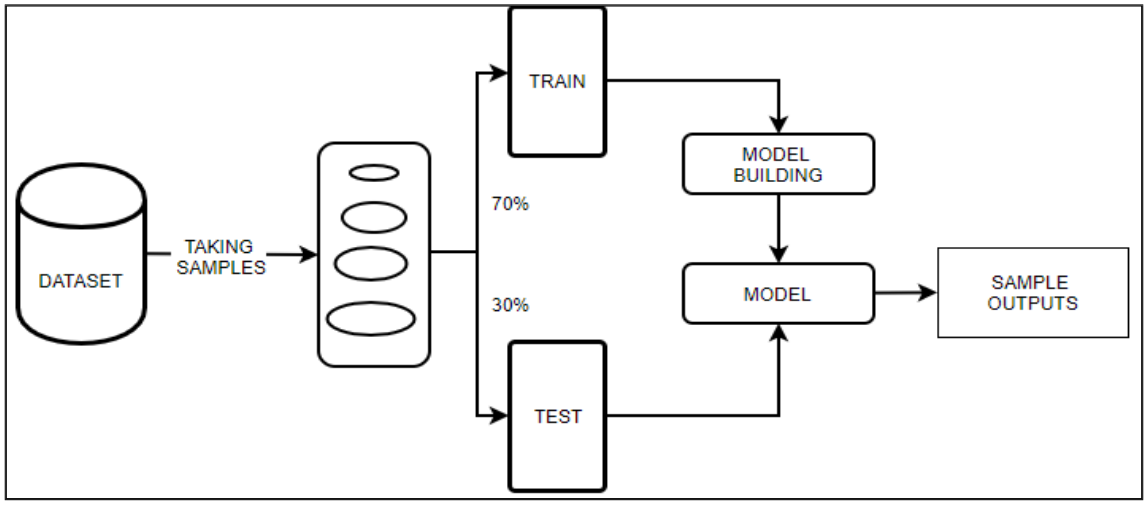

Figure 3: The overall pictorial description of building a sampling-based model with randomly segregated IDS forming different sample sizes.

4.3 The timing-based model and its performance

The general feeding of the entire data set as test and train sets to the model doesn't provide much understanding of the underlying functionality mostly in cases of the experiment where timing is a consideration. The deterioration of the disk happens by time (Zhao et al.; 2010) so, to predict the disk failure timing is to be considered as an important parameter. Each IDS recorded every hour has attributes of the disk which varies with timing. How well an algorithm could get in line with this variation of attributes contributing to failure, to predict the disk failure with minimum hours of data provided (when limited data is available) would state its performance. The performance of the algorithm is checked by varying the timing interval, a total of five different timing interval are considered.

To conduct this experiment the data set is partitioned into train and test set after randomly distributed them. This makes IDS from a single disk to be in both test and training set. The training sample which has several IDS from many disks is then grouped by the serial number (Disk-ID) of each disk. From this grouped disk "N" hours of IDS ("N" explicitly means timing in hours) from each disk are taken to build the model. Consider if N=8, the 8hrs of data from all the disk is used to train the model. The complete description of different "N" values that are taken and their results obtained is produced in Figure 9. Figure 4 illustrates the model building steps, taken to check the effect of timing on the model.

4.4 Applying voting to the predicted individual disk samples

All the above sections had a prediction where each IDS will have either *failed* or *good* as the result. But all the IDS predicted for a disk won't have the same output. The

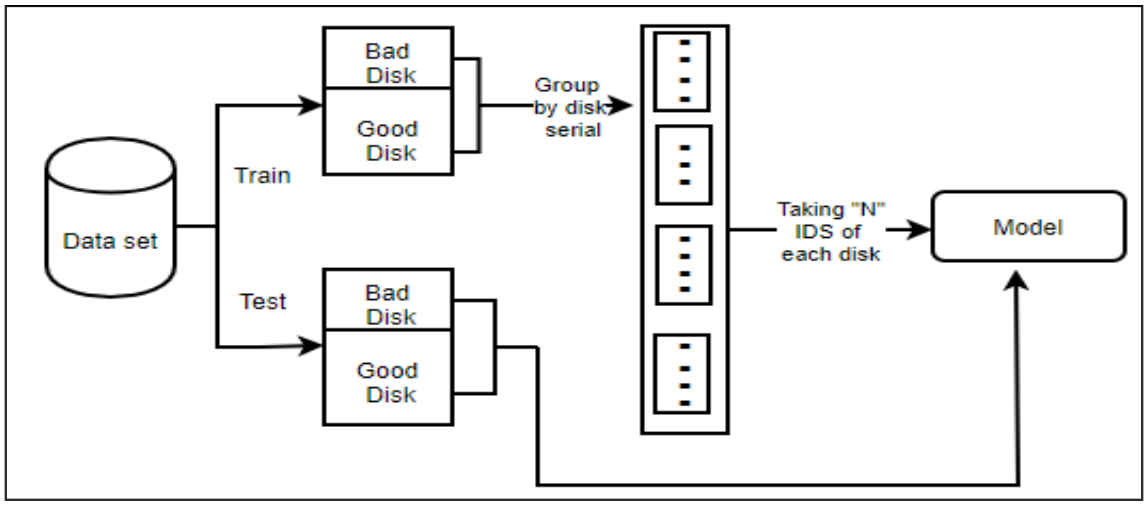

Figure 4: The description of steps taken to build a model with timed data. Here "N" hours of timed train data is used for training the model.

output could vary along the IDS of each disk when predicted by the model. A single disk IDS could have both the *failed* and *good* as the results. So, the status of the disk is not concluded by a single value, either *fail* or *good* but has IDS varying from *good* to *failed* and vice versa. In such a case, to determine the exact status of the disk a voting based method can be used which is expected to give a more accurate result. As it takes the majority votes into consideration while determining the status of the disk rather than a single IDS.

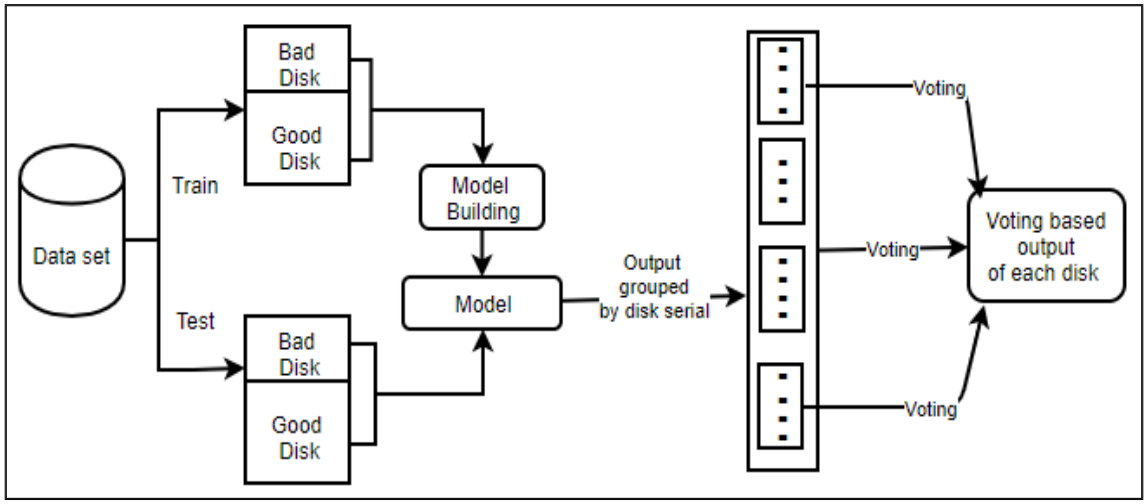

Figure 5: The pictorial description of voting being considered in the predicted results of the model to minimize misclassification of disk.

To conduct the voting based test on the algorithms the training data formed by combining the *good* and *failed* disk is used to build the models. The trained model is then tested in the testing set. The predictions are then grouped by the serial number given to the disk. This generates a group of IDS for each disk having both *failed* and

good as the result. Then "V" number of IDS predictions of each disk is taken to generate a voted output. In general, Once the value of "V" is confirmed, say V=10 the last 10 IDS of a disk is taken for voting to determine its status.

The value of "V" is varied and the lowest count of IDS for which the model attains the best result in classifying the disk properly into *good* and *failed* is noted, this would determine the performance of the different model. The minimum IDS the model takes to correctly define the status of the disk, the best the model performance is. The number of IDS the model correctly defines has an influence on the voting result. The description of the model performance for a different number of voting sample is given in Figure 10 and Figure 11. The illustration of the steps taken to build the model is given in Figure 5

4.5 Self-monitoring system

The SMART system works by maintaining thresholds on the attributes it collects from the hard drive. These SMART attributes are used by most of the researchers to build there machine learning algorithms to predict the disk failure. RF and other algorithm implemented in this paper also use the same SMART attribute as its data for building models and for testing. We used this SMART attributes to build a self-monitoring application that would track the health status of the inbuilt disk of the system in real time.

4.5.1 Crystal Disk Info

Crystal Disk Info (CDI) (*DiskInfo*; n.d.) is open source software that runs on a windows machine and is widely used to collect the SMART attributes of the internal disk in the system. A total of twenty-two attributes are collected by the software, this collection of the disk attribute can be timed. A frequency interval of five minutes is set for collection of the attributes which are then saved internally in a text file.

4.5.2 The h2o library in R

h2o.ai (Allen; 2004) is a fast scalable open source machine learning platform, providing support in Java, R, and python. It is available as an easily installable package in R. The RF is one among the many algorithms that are supported by h2o. The reason for choosing h2o is its capability to convert the machine learning model to java POJO class which is a machine learning implementation of open source framework of h2o.ai. After the conversion of R built a model to POJO it can be run in java by the *Gradle* a general purpose build tool.

4.5.3 Description of the Application

For implementing the application RF is used which is built in h2o with the sample of 800,000 rows from "W" dataset considered in 4.2. The reason for not considering timing or voting methods in building the system is because voting is used when several IDS is available as a result but in real time implementation, we get a single row of data with all the attribute, where voting is not possible. The timing-based is designed to check the modes performance when the data is limited but it's not the case here we have an ample data to train the model. The built model trained with a sample of data is then extracted as a java POJO class which is used as a backbone to build the application in

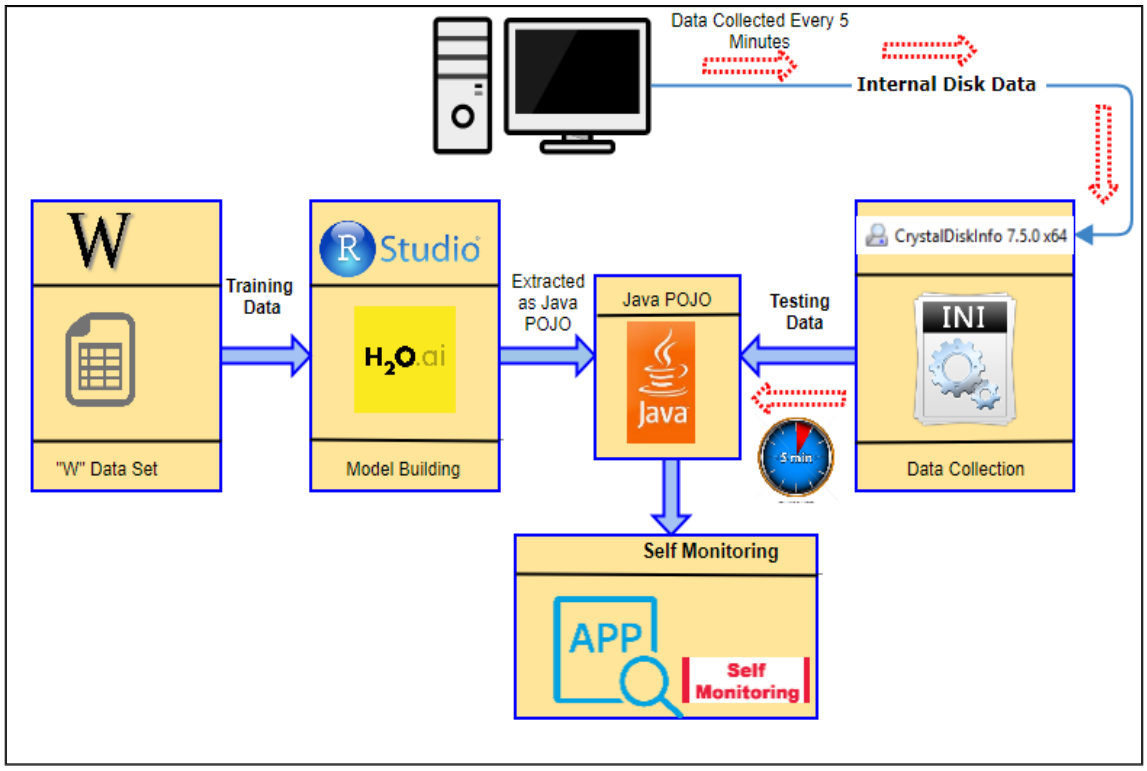

Figure 6: Overall illustration of SMS built on the most predictive model using h2o.

java. The internal disk data of the system is collected by the CDI and stored in a file which is processed and provided as a test sample to the model. The predicted status of the disk is given to its user. This internal disk data collection happens every 5 minutes and is tested with the model automatically and the status is updated dynamically. This entire system of operation is automated and its complete pictorial description is given in Figure 6. Figure 7 shows the flow of data and the internal working of SMS.

5 Evaluation

The implemented models in RStudio is evaluated under various conditions. As an evaluation metric, the TPR and False Positive Rate (FPR) were recorded for all the experiments conducted. The prediction of the failed disk where considered as a positive prediction, so the TPR is the number of failed disk correctly predicted as failed which is mentioned as FDR and the FPR is the good disk classified to be failed mentioned as FAR.

5.1 Performance measure with the variation in disk samples

To check the performance of the algorithm with different sample sizes, considerable samples of varied sizes were taken into consideration. The minimum number of sample considered was one hundred thousand and had a *failed* disk count of 417 and a *good* disk count of 22594 holding a *failed* samples of 3929 and *good* samples of 96072. The samples were then increased in such a way that the count of disk increased in the samples taken. The eight hundred thousand samples included IDS from all the *failed* and *good*

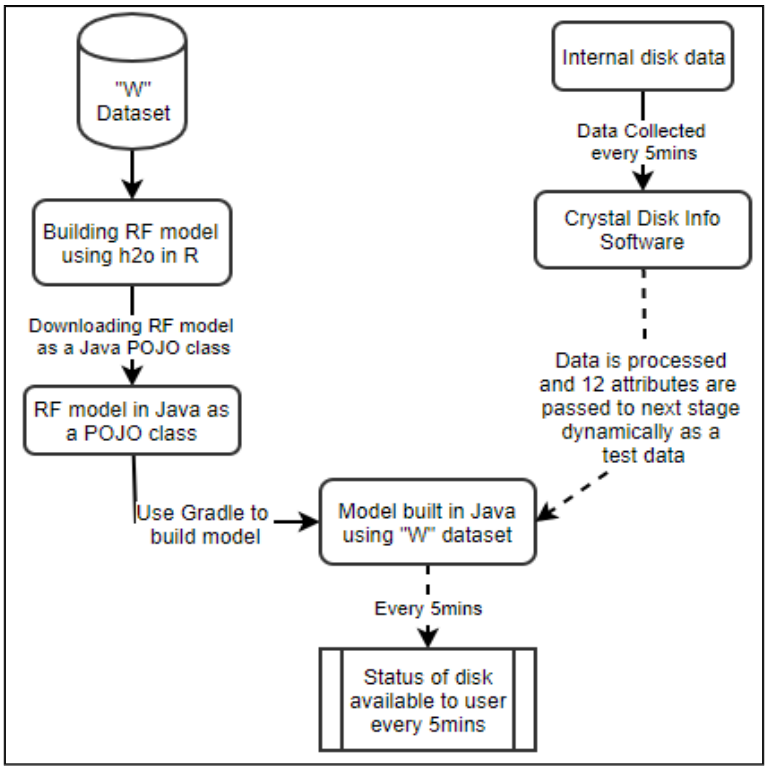

Figure 7: The flow diagram showing the in-depth concepts and functioning of SMS developed on RF model.

disk. The complete description of the recorded FAR and FDR for three algorithms is given in Figure 8.

The RF showed better prediction results with different sample size. Even with a smaller sample of one hundred thousand, it showed an FDR of 96% and FAR of 0.01% which is higher than the DT model built by Li et al. (2017) which showed a lower FDR of 82.3% at 0.09% FAR with a smaller sample. As the size of the sample increased RF showed a drastic improvement in performance with a higher FDR at a minimal FAR. The RF computes the importance of the variable with the trees it built internally. From the result obtained on the important Attributes *Power-On Hour* (POH) attribute followed by *Reallocated Sector* always remained as the most important with the *Pending Sector* as the least important attributes. With the 800,000 samples, the important attribute POH was used 52 times as a root node in the formation of internal trees with a maximum of 5585 child nodes beneath it. The interaction between the variables is assumed as an important parameter in Zhao et al. (2010), this interaction effect is considered by RF internally and it's seen that POH has maximum interacted with *Seek Error* nearly 88 times while building the model. The performance of the complete dataset was not recorded in the case of RF due to the lack of computational capability of the system used in building the model. As with larger dataset, the size of trees built internally increases, with the replica of data being held for each tree in the memory which leads to the consumption of a lot of memory. The memory consumption also increases linearly with the increase in the number of trees. Due to this memory limitation of RF with large dataset made it impossible to run on a smaller machine. The computation could have been made possible with a high-end machine, but its clear that the performance is

increasing with the increase in the sample size and its performance way better than DT algorithm with just a fraction of data.

	RF		K-means		FFN	
Samples	FAR	FDR	FAR	FDR	FAR	FDR
1000,00	0.00017	0.964	0.57	0.141	0.0012	0.886
2000,00	0.00028	0.971	0.57	0.162	0.0015	0.867
3000,00	0.0004	0.973	0.58	0.154	0.0022	0.765
4000,00	0.00021	0.978	0.571	0.133	0.0021	0.894
5000,00	0.00028	0.98	0.572	0.133	0.0014	0.87
6000,00	0.00018	0.981	0.572	0.135	0.0005	0.765
7000,00	0.00026	0.992	0.572	0.134	0.0026	0.851
8000,00	0.00017	0.992	0.573	0.134	0.0014	0.888
complete data set	-	-	0.427	0.866	0.0015	0.908

Figure 8: The disparity in the prediction results with variation of sample sizes

The FFNN showed a better performance than the k-means algorithm. It provided FDR of 88% with 0.1% FAR. With the increase in the sample size, the performance of FFNN also increased, with complete data set it showed 90.8% prediction at 0.15%. Similar results were obtained using Back Propagation on the same dataset by Li et al. (2017). The neural net mostly converged within hundred iterations so it's left unchanged throughout the experiment.

With the k-means clustering model, the FAR remained almost constant and higher than the FDR in most cases, showing an overall deficient performance. In some cases, it showed a drastic improvement in FDR. The k-means clustering provides greater deviation and different results while running the same samples this is due to the selection of different starting point by the algorithm. The initial partition has a greater effect and results in different final clusters. This was helped by setting a seed which regained the same best result obtained by k-means for a sample. But as a whole, this algorithm lacked it's performance when compared to the result obtained by Li et al. (2017) and it was also lower than the one obtained by the SMART algorithm (Allen; 2004).

5.2 Timing consideration in the training data

This experiment provides an idea about the performance of the model when provided with "N' number of IDS samples before its actual status. The IDS are recorded at a frequency of one hour. The timing in hour indirectly refers to the "N' number IDS before the actual disk status.

Five different timing in hours where considered which are 12hr, 24hr, 48hr, 96hr, and 169hr. For each of the timing interval, the maximum FDR with minimum FAR was recorded. To provide an accurate result the average of ten runs for each timing is taken into consideration for evaluating the models. The RF obtained the best result when the timing was set to 169hrs providing FDR of 99% and FAR of 0.02%. With the decrease in the timing the data considered in the training of the model decreases which leads to the decreased prediction capability of the model, it also occurred in the previous section where different samples were considered.

The FFNN model built with twenty-four hours of data showed a better prediction

	RF				FFNN				k-means			
Timing in hrs	Max.FDR	Min.FAR	Avg.FDR	Avg.FAR	Max.FDR	Min.FAR	Avg.FDR	Avg.FAR	Max.FDR	Min.FAR	Avg.FDR	Avg.FAR
12	0.9844	0.00006	0.98	0.000253	0.89432	0.00048	0.862	0.00047	0.913	0.743	0.087	0.427
24	0.9888	0.00016	0.986	0.00021	0.8979	0.0011	0.879	0.00045	0.6159	0.425	0.062	0.427
48	0.9912	0.0002	0.989	0.00022	0.90278	0.00266	0.882	0.0019	1	0.0025	0.991	0.875
96	0.9941	0.0002	0.991	0.000236	0.9269	0.0012	0.894	0.00155	1	0.0025	0.886	0.068
169	0.9957	0.0003	0.992	0.000237	0.9194	0.0013	0.864	0.0015	1	0.0025	0.968	0.371

Figure 9: The performance measure of the model recorded with different timed training data.

result, with 87% FDR at 0.04% FAR. Although it shows a considerable performance its lower when compared to RF. The k-means clustering which selects random starting point showed 88% FDR with 6.8% FAR which is lesser than other algorithms. The maximum value of 100% FDR was recorded in many cases by k-means. The stability of the algorithm remains so low that it shows a greater deviation between the maximum and average reading. The complete representation of models performance when evaluated by different timing interval is shown in Figure 9

5.3 The voting on predicted individual disk samples

All the above sections had predicted for IDS to have either *failed* or *good* as the result. But this output could vary along the IDS of each disk when predicted by the model. So, a single disk has many IDS which will have both the *failed* and *good* output as the results predicted by the model. In this case to determine the exact status of a disk voting based method is used which is expected to give the more accurate result. As it takes the majority votes of IDS results into consideration while determining the status of the disk rather than a single last IDS predicted by the model. The voting introduced in the results lowers the misclassification of the disk and provides a positive impact on the predicted result. The Figure 10 and Figure 11 provides the misclassification of the good

RF		
No. of IDS considered for voting	No. Misclassified failed disk	No.of missclassified good disk
3	1	2
10	0	2
27	0	2
50	0	1

Figure 10: The variation in disk misclassification when voting is introduced to the RF predicted results.

and failed disk and its reduction when the voting is introduced. Although the tabulated result shows a single unit variation in terms of misclassification of the disk, it considers

several samples to determine this status. The FFNN showed better improvement when

FFNN		
No. of IDS considered for voting	No. Misclassified failed disk	No.of missclassified good disk
3	2	12
10	2	11
27	1	11
50	1	11
60	0	11
100	0	11

Figure 11: The variation in disk misclassification when voting is introduced to the FFNN predicted results.

the voting method is introduced to its samples of output. But with almost hundred samples of each disk being considered for voting, there was not much variation in the misclassification rate of the disk. It is obvious that the algorithm itself didn't provide much accuracy in classification of the IDS to their specific category of *good* and *failed*.

In case of RF, it's clear that the algorithm performance is way better than FFNN as disk classification is concerned. With the minimum of three sample it provided better classification and with an increase in the sample to fifty it showed zero classification error for the *failed* disk with a single disk misclassified in case of *good* ones. The RF has its own voting considered on the tree it builds internally so its performance is Superior to other algorithm considered. The internal trees built be the RF shows that with 40 trees the error in classification subsides below 0.01 with the increase in the tree to 100 it lowers to a range near 0.008 which shows the supremacy of RF.

The overall look of the complete statistical description of the best results obtained under different condition by each algorithm compared with the DT is given in table Figure 12.

Algorithms	Samples of data			Timing window			voting method		
	Sample size	FDR%	FAR%	Timing	FDR%	FAR%	Votes Considered	FDR%	FAR%
DT	900,000(Apx.)	96.9	0.2	169hrs	95.5	0.09	27	93	0.009
RF	800,000	99	0.01	169hrs	99	0.02	3	99	0.005
FFNN	100,000	88	0.12	96hrs	89.4	0.15	50	94	0.05
K-means	400,000	57	13	96hrs	88.6	6.8	-	-	-

Figure 12: The overall statistical results showing the best performance of the algorithm under different condition in comparison with the DT results.

5.4 Run time estimates

The RF model on which the SMS application is built has FDR of 99% with 0.01% FAR. The maximum time taken to build the model in Gradle is 1.26mins. After a single

build its tested with updated disk data every 5mins to check the status of the disk. Among the algorithms, the K-means took the least time of 22.39secs to build the model with a sample size of 800,000 rows. The FFNN and RF showed 5.16mins and 3.57mins under the same condition. Figure 13 shows the run time estimates of the algorithms.

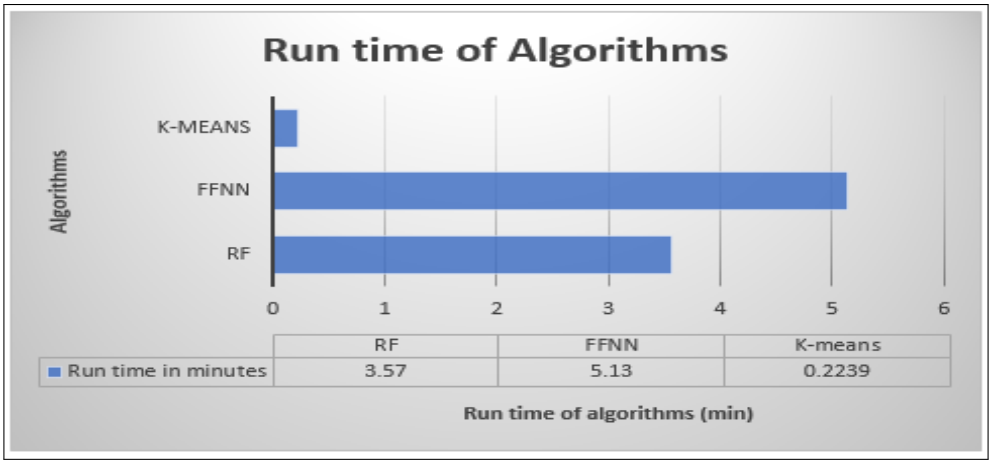

Figure 13: The time taken by the algorithms to build the models with a train set of eight hundred thousand rows.

5.5 Discussion

The findings from this study showed an improvement over the existing prediction rate of the disk with reduced FAR with the model built using RF. The RF has a steady performance when put to test with different samples of data, with a maximum of 3% difference with a wide range of samples. The performance of RF is also superior when tested with timing and voting methods. It is made possible to predict the disk status 24hr before the failure with a FAR of 0.01% by a model built on RF. With this credibility, a real-time prediction of underlying disk status using the RF mode is implemented.

Surprising results were observed with the introduction of under-sampling to balance the data set for prediction. The results provided showed an increase in the FAR for both the RF and the FFNN with a moderate increase in the prediction rate for FFNN. This is due to the increase in the minority (failed disk samples) samples in the test set after performing under-sampling. Similarly, when the voting procedure is used on the predicted IDS of all the three algorithms together to determine the status of a disk, the number of disks properly classified decreased. The reason behind this effect is apart from RF the K-means and FFNN wrongly classify most of the IDS. So, while using voting on the three the result from K-means and FFNN is picked as the major vote which leads to the wrong classification. Some of the limitation came from the computational incapability to check the RF model on the entire dataset.

6 Conclusion and Future Work

In this paper, we propose models based on RF, FFNN, and k- means clustering for building a stable and more accurate predictive model with maximum FDR at minimum

FAR. The comparison of our models with states of the art model such as DT is done under various conditions of sampling, timing and voting showed that the RF performs best by predicting the maximum failing disk with a reduced FAR. The RF showed a stable prediction even with a minimal sample size when compared to DT. In particular, RF model attained a maximum prediction of 99% with a FAR of 0.01% whereas DT showed only 93% FDR with 0.01% FAR.

For building and evaluation of the model, we considered real-world dataset with many *good* and *failed* disks. To extract the practical benefit from the RF model, we built a real-time predicting system using RF to monitor the status of the underlying disk which provides a prediction rate of 99% which is far superior to the inbuilt SMART system which has only 3% to 10% prediction capability. As a future work, the implemented prediction system could be extended for predicting disk status from a wide range of disks. And we believe the RF algorithm provides stable and good failure prediction with HDD its worthwhile to check RF on a wider range of dataset with several classes of disk drives.

References

Allen, B. (2004). Monitoring hard disks with smart, *Linux Journal* (117).

Bartlett, E. B. and Uhrig, R. E. (1992). Nuclear power plant status diagnostics using an artificial neural network, *Nuclear Technology* **97**(3): 272–281.

Botezatu, M. M., Giurgiu, I., Bogojeska, J. and Wiesmann, D. (2016). Predicting disk replacement towards reliable data centers, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, pp. 39–48.

Breiman, L. (2001). Random forests, *Machine learning* **45**(1): 5–32.

Chaves, I. C., de Paula, M. R. P., Leite, L. G., Queiroz, L. P., Gomes, J. P. P. and Machado, J. C. (2016). Banhfap: A bayesian network based failure prediction approach for hard disk drives, *Intelligent Systems (BRACIS), 2016 5th Brazilian Conference on*, IEEE, pp. 427–432.

Díaz-Uriarte, R. and De Andres, S. A. (2006). Gene selection and classification of microarray data using random forest, *BMC bioinformatics* **7**(1): 3.

DiskInfo (n.d.). https://crystalmark.info/software/CrystalDiskInfo/index-e.html.

Gehrer, M. (2016). Predicting disk failures for reliable clouds.
URL: *https://www.ibm.com/blogs/research/2016/08/predicting-disk-failures-reliable-clouds/*

Hamerly, G., Elkan, C. et al. (2001). Bayesian approaches to failure prediction for disk drives, *ICML*, Vol. 1, pp. 202–209.

Hartigan, J. A. and Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm, *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **28**(1): 100–108.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks, *Neural networks* **4**(2): 251–257.

Hornik, K., Stinchcombe, M. and White, H. (1989). Multilayer feedforward networks are universal approximators, *Neural networks* **2**(5): 359–366.

Hoskins, J., Kaliyur, K. and Himmelblau, D. (1990). Incipient fault detection and diagnosis using artificial neural networks, *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*, IEEE, pp. 81–86.

Hughes, G. F., Murray, J. F., Kreutz-Delgado, K. and Elkan, C. (2002). Improved disk-drive failure warnings, *IEEE Transactions on Reliability* **51**(3): 350–357.

Kaplan, J. (2008). Meeting the demand for data storage.
URL: *http://www.computerweekly.com/feature/Meeting-the-demand-for-data-storage*

Khalilia, M., Chakraborty, S. and Popescu, M. (2011). Predicting disease risks from highly imbalanced data using random forest, *BMC medical informatics and decision making* **11**(1): 51.

Kirkos, E., Spathis, C. and Manolopoulos, Y. (2007). Data mining techniques for the detection of fraudulent financial statements, *Expert systems with applications* **32**(4): 995–1003.

Leshno, M. and Spector, Y. (1996). Neural network prediction analysis: The bankruptcy case, *Neurocomputing* **10**(2): 125–147.

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

Li, J., Stones, R. J., Wang, G., Liu, X., Li, Z. and Xu, M. (2017). Hard drive failure prediction using decision trees, *Reliability Engineering & System Safety* **164**: 55–65.

Li, Z., Li, Y. and Xu, L. (2011). Anomaly intrusion detection method based on k-means clustering algorithm with particle swarm optimization, *Information Technology, Computer Engineering and Management Sciences (ICM), 2011 International Conference on*, Vol. 2, IEEE, pp. 157–161.

Lima, M. F., Zarpelao, B. B., Sampaio, L. D., Rodrigues, J. J., Abrao, T. and Proença, M. L. (2010). Anomaly detection using baseline and k-means clustering, *Software, Telecommunications and Computer Networks (SoftCOM), 2010 International Conference on*, IEEE, pp. 305–309.

Limpon, R. (1987). An introduction to computing with neural nets, *Ieee assp magazine* pp. 4–22.

Murray, J. F., Hughes, G. F. and Kreutz-Delgado, K. (2003). Hard drive failure prediction using non-parametric statistical methods, *Proceedings of ICANN/ICONIP*.

Murray, J. F., Hughes, G. F. and Kreutz-Delgado, K. (2005). Machine learning methods for predicting failures in hard drives: A multiple-instance application, *Journal of Machine Learning Research* **6**(May): 783–816.

Peng, D. (2017). Global shipments of hard disk drives.
URL: *https://www.statista.com/statistics/275336/global-shipment-figures-for-hard-disk-drives-from-4th-quarter-2010/*

Phua, C., Lee, V., Smith, K. and Gayler, R. (2010). A comprehensive survey of data mining-based fraud detection research, *arXiv preprint arXiv:1009.6119* .

Queiroz, L. P., Rodrigues, F. C. M., Gomes, J. P. P., Brito, F. T., Brito, I. C. and Machado, J. C. (2016). Fault detection in hard disk drives based on mixture of gaussians, *Intelligent Systems (BRACIS), 2016 5th Brazilian Conference on*, IEEE, pp. 145–150.

Saeys, Y., Inza, I. and Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics, *bioinformatics* **23**(19): 2507–2517.

Tam, K. Y. and Kiang, M. Y. (1992). Managerial applications of neural networks: the case of bank failure predictions, *Management science* **38**(7): 926–947.

Wang, Y., Ma, E. W., Chow, T. W. and Tsui, K.-L. (2014). A two-step parametric method for failure prediction in hard disk drives, *IEEE Transactions on industrial informatics* **10**(1): 419–430.

Wang, Y., Miao, Q. and Pecht, M. (2011). Health monitoring of hard disk drive based on mahalanobis distance, *Prognostics and System Health Management Conference (PHM-Shenzhen), 2011*, IEEE, pp. 1–8.

Wang, Y., Tsui, K., Ma, E. W. and Pecht, M. (2012). A fusion approach for anomaly detection in hard disk drives, *Prognostics and System Health Management (PHM), 2012 IEEE Conference on*, IEEE, pp. 1–5.

Wei, C.-P. and Chiu, I.-T. (2002). Turning telecommunications call details to churn prediction: a data mining approach, *Expert systems with applications* **23**(2): 103–112.

Yuan, G., Li, B., Yao, Y. and Zhang, S. (2017). A deep learning enabled subspace spectral ensemble clustering approach for web anomaly detection, *Neural Networks (IJCNN), 2017 International Joint Conference on*, IEEE, pp. 3896–3903.

Zhao, Y., Liu, X., Gan, S. and Zheng, W. (2010). Predicting disk failures with hmm-and hsmm-based approaches., *ICDM*, Springer, pp. 390–404.