International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Utilization of Genetic Algorithm and Significance Scores for Feature Selection in the Interest of Increasing Accuracy of Fault Detection in Hard Disk Drives for HDFS

[1] B. K. Prasad Banavathu, [2] A. Ananda Rao

Submitted: 26/05/2023 **Revised**: 07/07/2023 **Accepted**: 25/07/2023

Abstract— The term "hard disk drive" (HDD) refers to a storage device used in computers and servers. If these components suddenly stop working, vital information could be lost forever. Most hard disk drives (HDD) include SMART technology, which allows them to track a variety of performance metrics and report on their own health status. However, not all SMART characteristics may be relied upon to spot a failing HDD. In this research, we offer a two-stage process for choosing the best HDD failure indicators. First, a GA is used to narrow down the SMART qualities to a manageable set that yields feature vectors that are intuitive to separate and naturally cluster. The best subset of features is determined by the GA based solely on the fitness of a set of SMART attribute pairs. The use of a significance score to measure a feature's statistical impact to disk failures in a second layer is suggested to improve the GA's feature selection even more. This hand-picked collection of SMART traits is used to train the naive Bayes classifier, a generative classifier. The suggested approach outperforms cutting-edge alternatives in terms of failure detection and false alarm rate, according to extensive testing on a SMART dataset obtained from a commercial datacentre. There is no need to fine-tune any parameters or thresholds, and the classifier just needs to be trained on a smaller set of SMART properties.

Keywords— *Genetic Algorithm, Significance Scores, Fault Detection in HDD, SMART.*

I. Introduction

The term "big data" is used to describe data sets that contain an enormous amount of information, much exceeding the capacity of traditional data management and analysis systems. The actual challenge is finding or creating the most reliable means of extracting value from the massive volumes of scalable data that are being collected as technology develops. In order to analyze massive and heterogeneous data sets, "big data" analytics necessitates gathering information from numerous sources. It typically involves the execution of numerous separate analytics algorithms, which necessitates access to high-performance computer resources and the capacity to integrate massive and diverse data sources. The main challenge is carrying out the many separate analytics that make up larger data and model work processes, as well as having access to very complicated amounts of data. Data analytics technology and methods enable researchers to examine large datasets and draw

meaningful findings. Distributed computing on multiple servers, dynamic workload balancing, data integration, high availability, and job prioritization are all essential to adjusting to and tolerating Big data technology's importance in the modern world.

Hadoop is a java-based open-source framework developed by Apache that enables distributed processing of large datasets across clusters of computers with minimal need for complex programming models.

Figure1 shows the overall framework for big data analytics that makes use of the Hadoop idea. Data from both internal and external frameworks is routinely incorporated into big data analytics applications, with Hadoop acting as the major repository for raw data streams in order to handle massive amounts of data on huge clusters in a dependable manner. In the end, data should be coordinated and designed to gain exceptional performance in distributed applications once it has been prepared, evaluated using the analytical processing software's, and saved in the Hadoop Distributed File System (HDFS) [16]. Apache Hadoop was designed to work with both structured and unstructured data in a cycle. HDFS and MapReduce make up Hadoop. HDFS helps archive huge datasets. Large informative indexes and rapid data access make this sector ideal for applications with enormous datasets.

[1]*Research Scholar, Computer Science and Engineering, Jawaharlal Nehru Technological University Anantapur (JNTUA), Ananthapuramu, Andhra Pradesh 515002, India,* bcogbi@gmail.com.
[2]*Professor, Computer Science and Engineering, Rayalaseema University, Kurnool, Andhra Pradesh 518001, India,* akepogu@gmail.com.

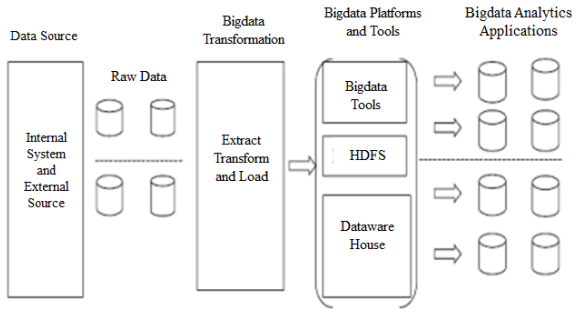

Fig 1: Analytical Framework for Big Data Sets

HDFS's MapReduce programming method parallelizes and distributes massive datasets [12]. The framework's efficiency and output will improve with big data inquiry. Big Data analytics works well with SSDs [9], which store data in multiple locations. SSDs are the best way to boost I/O and throughput since they have no single point of failure [2] and allow linear I/O scaling. They scale easily to any performance and capacity. HDFS's write, delete, and disk replacement [13] procedures may create drive data placement [14] discrepancies. This makes Data Nodes very skewed.

MapReduce processes Hadoop. MapReduce breaks down large datasets into smaller, manageable parts for parallel processing. Hadoop MapReduce users define the map and reduce functions, and HDFS is usually utilized, therefore task I/O performance [16] can depend on HDFS. Hadoop MapReduce processes "key-value pairs," hence its name. For each pair, call the mapper function. Hadoop MapReduce then categorizes map-phase pairs using k. Map jobs save results locally, not to HDFS. Parsing the map function's output invokes the reducer function for each key k and value l. MapReduce processes data, while HDFS stores it. Hadoop uses HDFS. It can store and share lots of data across a network. HDFS has many Data Nodes for storage and one NameNode for metadata and monitoring. Programmers created the MapReduce programming concept and implementation to process huge data volumes.

Cloud computing and virtualization have made data centers important to Internet service providers' network design. This system requires secure, continuous data storage. If an HDD fails, client data, transaction records, and sales numbers may be lost. Third-party data puts the hard drive failure rate at 14%, despite industry figures of 1%. Predictive HDD maintenance is crucial because disk failure [6] usually occurs without warning. "Preventive Maintenance" [4] reduces equipment breakdowns via a time- or meter-based maintenance program. Managers may be fooled by preventative maintenance techniques that don't reflect device conditions. Predictive maintenance (PM) saves money and extends machine life by studying past data to predict machine faults [7].

Precursors can indicate a failure's approaching occurrence. Precursor parameter [8] changes can predict failure, which can be prepared for. Identifying [3][5] and monitoring precursor parameters. This data can help qualify products and predict issues. Zhang discussed power supply Prognostics and Health Management (PHM). After establishing a baseline, he analysed historical data to uncover precursor parameters. He then identified precursor characteristics for a switch-mode power supply. PM emphasizes direct equipment performance monitoring during regular operation to predict failure. Instead of waiting a predetermined number of hours before scheduling [18] repair regardless of performance, equipment can be monitored for signals of approaching failure by collecting data on vibration, temperature, and other characteristics. This beats scheduling [17] maintenance after a certain amount of hours. Big data analysis can handle enormous amounts of data from several sources. Preventative maintenance can efficiently diagnose and fix many devices while reducing costly repairs. Big data methods allow PM to continuously gather and analyze data to identify patterns that could improve device performance. Edge recommended fraud management and prevention architecture based on best practices. This architecture can detect fraud and suspicious behaviour in real data flows and block fraudulent transactions. Ko advocated using whole time series data streams to detect and monitor intrusions. Yang used Bayesian robust principal component analysis (RPCA) to identify road traffic occurrences. This incremental data stream analysis can detect online real-time occurrences at cheap computational cost.

II. Related Work

It is possible to transfer data quickly across different computing nodes while using HDFS. It was first linked with MapReduce, which is a data processing framework that filters input, distributes jobs across a cluster's nodes, and aggregates the results into a coherent response to a query. This was done when the system was in its early phases. In a similar manner, when HDFS receives data, it breaks the data up into blocks and then distributes those blocks among all of the nodes in a cluster. When using HDFS, data only needs to be entered into the server once, but it may be retrieved and utilized in a variety of ways over and over again. When it comes to keeping track of the storage nodes in the cluster on which individual files are kept, HDFS is dependent on a centralized NameNode. There is another kind of HDFS node called a DataNode, and it can be found in a cluster that consists of commodity hardware. The DataNodes for a given network are often co-located in the same server rack. After then, distinct portions of the data are distributed across the many DataNodes. Blocks are duplicated across nodes in order to promote effective parallel processing as much as possible. The NameNode is aware of both the DataNodes and their positions within the machine cluster. The

NameNode is responsible for managing not only the replication of data blocks between DataNodes but also other file operations. In order for the NameNode to perform its purpose, the DataNodes must be located in close proximity to it. This indicates that the cluster is able to add or remove nodes on the fly in order to accommodate varying demands for the capacity of the servers. DataNodes maintain consistent communication with the NameNode in order to establish whether or not they are responsible for the completion of any outstanding tasks. As a direct consequence of this, the NameNode possesses unrestricted insight into the state of all DataNodes. If the NameNode determines that one of the DataNodes is failing to perform its duties, it is able to transfer those responsibilities to another node in the network that possesses the same data block. Due to the fact that DataNodes are able to communicate with one another, they are able to cooperate during routine file operations. The HDFS was designed with redundancy and the ability to tolerate errors in mind throughout construction. The file system makes a copy of each piece of information and then distributes those copies to several nodes, with at least one copy being kept in a server rack that is physically separate from the others.

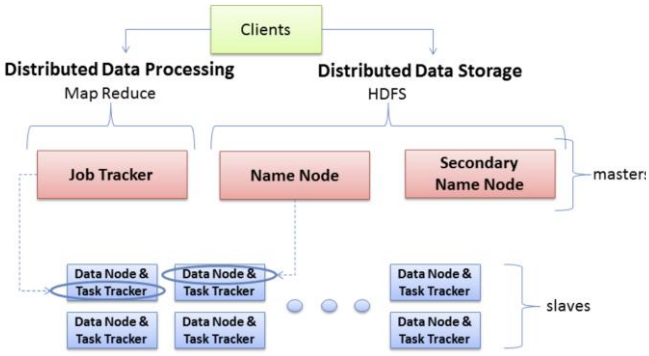

Fig 2: HDFS file system working process from client side

HDFS employs a master-slave design. The major server that is in charge of the file system namespace and managing client access to files within an HDFS cluster is referred to as the NameNode. Hadoop's distributed file system has a component called the NameNode at its core. This component is in charge of providing clients the required permissions and managing the namespace for the file system. The DataNodes in the system are in charge of monitoring all of the storage devices that are linked to the nodes that they are operating on. Users are able to store their data in files thanks to the fact that HDFS exposes a file system namespace to its users. A single file can be broken down into numerous chunks, each of which is stored in its own distinct DataNode. Within the context of the file system namespace, the NameNode is responsible for a variety of tasks, including the opening and closing of files and directories, as well as their renaming. The mapping of blocks to data nodes is another function controlled by the NameNode. The DataNodes are responsible for processing the requests made by the users of the file system to read from or write to the system. In addition, they are able to generate, delete, and replicate blocks in response to commands sent by the NameNode. Standard hierarchical file formats may be utilized with HDFS because of this feature. A user or an application themselves can create directories to organize files in a way that is more manageable for both parties. Users are granted the ability to add, delete, rename, and relocate files inside the namespace hierarchy of the file system, just as they would with any other file system. The NameNode will make a note of any changes that occur to the namespace of a file system or the attributes of the system whenever they occur. The number of copies of a file that HDFS needs to store can be specified by an application. The replication factor of a file, often known as the total number of times it has been duplicated, is something that the NameNode keeps track of.

III. Methodology

Figure 3 depicts the proposed approach for determining whether or not hard disks are failing to function properly. It examines the SMART [10] properties of the disk to evaluate whether or not the disk is in good health. The contemporary HDD will acquire SMART properties in order to perform its own internal monitoring. The number of SMART qualities selected determines the exact dimensions of the high-dimensional space in which these traits exist. Drives in good condition and those that are failing are expected to cluster in the high-dimensional feature space in various ways, making them easy to identify individually. Disk failure may be predicted using SMART attributes if a classifier could be taught to distinguish between the two groups. This would negatively impact the classifier's performance because not all SMART attributes would result in equally useful separation of features. Therefore, feature selection is employed to determine which subset of characteristics best produces clusters of feature vectors that are both compact and easily discernible. To do this, one must first choose which characteristics will be used, and then choose the best subset of features to use for those features.

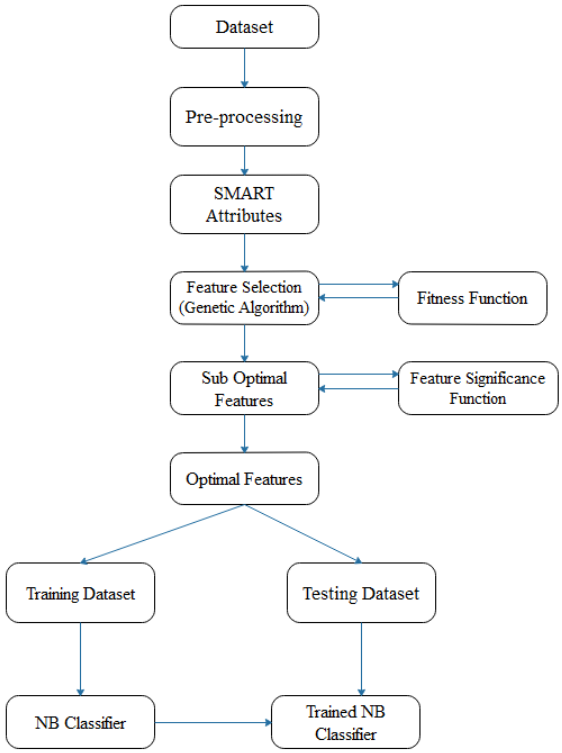

Fig 3: The suggested approach uses a feature significance function and a genetic algorithm to identify malfunctioning hard disks

In addition to enhancing the classifier's ability to make accurate predictions, selecting appropriate features can cut down on the amount of data that must be measured and stored, as well as the amount of time needed for both training and prediction. The goal of this study is to present a two-stage procedure for selection of features that employs a GA and a significance function for features. Each candidate feature is given a score of importance by the feature significance function, and the GA then assesses several feature combinations to find the optimal one. It throws out the features that have been analysed and found to have a statistically insignificant impact on the likelihood of a disk failing. After the final set of features has been selected, a Bayes classifier that can identify a failing HDD is trained using those features.

A. Feature Selection

In order to differentiate one thing from another using machine learning techniques, models of labeled objects are constructed using these techniques. Feature vectors are what are used to describe these things. The quality of these features is directly related to the accuracy of these models; To be more precise, a machine learning model's accuracy increases as feature discrimination improves. However, not all of an object's defining characteristics will be helpful in distinguishing it from other objects of the same type. If unnecessary features are prioritized during model construction, it could slow down the machine learning process and reduce the model's classification accuracy. Therefore, feature selection algorithms play an important role in many applications of machine learning. This is because these algorithms aid in the elimination of superfluous or redundant features, which in turn boosts classification accuracy, reduces the time needed to build the model, and cuts down on the number of training it takes to attain greater generalization.

The following three categories are the broad divisions that may be made for feature selection methods:

- **Filter Based Methods:** These approaches rank a set of characteristics using a fitness function, and then pick the features with fitness function scores that are above a threshold. The foundation of these techniques is the removal of irrelevant characteristics before moving on to the creation of the machine-learning model. The quality of the fitness function utilized has a major impact on how effective this method is. Filter-based methods like the one used to pick features for this inquiry are one type of such technique.

- **Wrapper Based Methods:** When utilizing this approach to train an ML model, bad data is not first filtered away. Instead, they rely on the classifier to sift through data and eliminate superfluous items. For instance, the classifier may use a number of different feature combinations, ultimately settling on the one that yields the maximum classification accuracy. This strategy can be laborious and isn't guaranteed to always be effective.

- **Embedded or Hybrid Methods:** These methodologies have either an embedded or a hybrid approach, as their names imply. These methods are faster because, unlike wrapper-based alternatives, they do not rely on the classifier's iterative usage to finish their work. Wrapper-based methods, on the other hand, cycle through various feature combinations and choose the optimal features subset depending on the accuracy of the classifier. Similarly, these techniques, unlike filter-based ones, do not employ a unique fitness function to rank the numerous qualities. Instead, these strategies might make use of the results produced by the classifier in order to choose the optimal subset of characteristics. In logistic regression and neural networks, for instance, the inputs (features) are ranked and the optimal subset is selected based on the weights assigned to them.

B. Genetic Algorithm based feature selection

Biological evolution and natural selection serve as the conceptual bedrock upon which a GA is built. It is generally accepted that the processes involved in natural selection are responsible for the development of living species. Over the course of their existence, living beings gradually acquire the features that will allow them to flourish despite the challenges posed by their surroundings. A GA accomplishes a similar goal by iteratively refining a given solution through the increasing selection of better candidate solutions and the discarding of choices that are deemed to be less desirable. By

doing so, it imitates the processes that drive the development of biological species, notably crossover and mutation. In order to get an accurate assessment of the quality of each solution, a fitness function or an objective function is utilized. As shown in the following example, a GA takes as input a collection of vectors in the space Rm with m dimensions.

$$X1(m),X2(m),....Xk(m), \quad (1)$$

Here

$$Xt(m)=[x1,x2,....xm-1,xm] \quad (2)$$

The GA generates an n-dimensional subset of Rn vectors using Equation (3):

$$X1(n),X2(n),.....Xk(n), \quad (3)$$

Where

$$Xt(n)=[x1,x2,....xn-1,xn] \quad (4)$$

The only change that the GA does is to lower the number of dimensions of each vector in the set described by Equation (1), but it does not change the cardinality of the set, which remains the same at nm. The dimensions that have been chosen by the GA are, for the most part, those SMART qualities that have the effect of reducing the fitness function. Therefore, X(n)i represents a vector that was picked by the GA to be part of the optimal set of features to be selected. The proposed fitness function in this work is shown in Equation (5).

$$F=CS \quad (5)$$

As stated in Equation (6), the average compactness of the classes is denoted by the letter C in this context.

$$C=1L1LCt \quad (6)$$

$$S=2L(L-1)t\neq fLStf \quad (7)$$

Figure 4 depicts a three-dimensional representation of the concepts of compactness and separability of two classes. These concepts are discussed in relation to a class. The degree to which distinct instances of a given class are grouped together is referred to as the compactness of that class. However, the ease with which clusters formed by examples from different classes can be separated is quantified by measuring the separability between the classes.

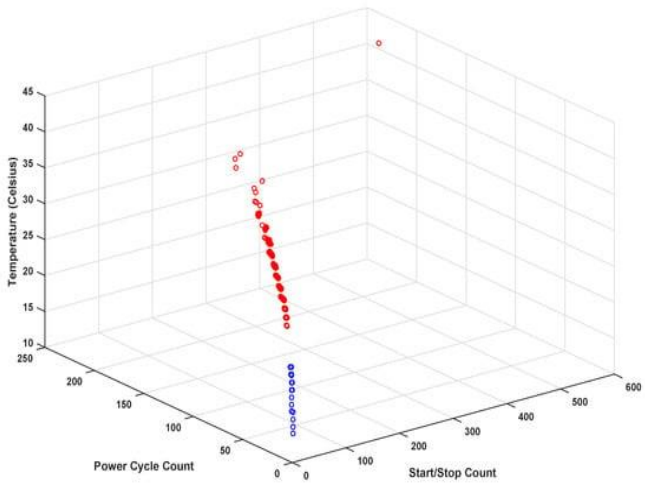

Fig 4: Healthy (blue) and failing (red) HDDs' three SMART attributes.

The concept of the Fisher ratio is conceptually comparable to the fitness function that has been presented, and in some research, the Fisher ratio has been utilized in conjunction with the proposed fitness function in order to pick features. The fitness function, expressed by Equation (5) as a ratio of the typical values of these two numbers, is minimized through a series of iterations by the genetic algorithm [11]. Figure 5 depicts the proposed two-tier method for feature selection, which makes use of a GA as well as scores indicating the relevance of the features. The SMART features or attributes must be chromosomally encoded in order to pass the GA. Through the process of mutations and chromosome crossovers, new chromosomes can be created from previously existing ones. The newly created chromosomes take the place of their parents if, and only if, they are superior to their parents, which means that they have a higher fitness or objective function performance than their parents did. This was covered in the previous section. As long as the fitness function is still increasing, the process of producing a new set of chromosomes and selecting the best ones to replace the old ones can be repeated indefinitely.

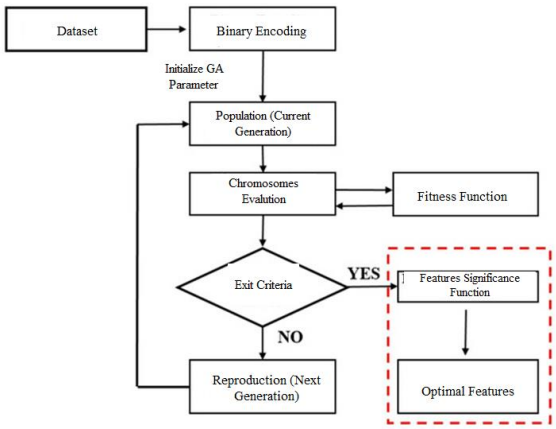

Fig 5: The genetic algorithm and feature significance scores-based feature selection algorithm's direction

A binary encoding approach is utilized in order to impart the SMART characteristics into the chromosomes. The resulting chromosomes are just lists of ones and zeroes, with zeros denoting that a certain SMART trait was not chosen and ones indicating that it was. We choose binary encoding because we require the flexibility to selectively add or exclude features. Only when a particular feature has been chosen will it be factored into the computation of the fitness function. On the other hand, if a certain characteristic is not chosen, then the calculation of the fitness function will proceed without taking into account that feature at all. Due to our lack of interest in determining feature weights, we have opted for the binary encoding strategy rather than, example, the value encoding technique. Instead, we're interested in identifying the minimal collection of traits necessary to achieve the desired fitness level. This is why the binary encoding scheme is recommended. The indices of each 1 and 0 in each chromosome are used to uniquely identify each of the distinct SMART characteristics. The initial population for the Genetic Algorithm is comprised of a string of random ones and zeros (1s and 0s), more specifically, a random selection of SMART characteristics. By causing these chromosomes to mutate and cross over with one another, new populations can be derived from the same set of genetic material. A crossover is the process of exchanging information or pieces between two parental chromosomes at places chosen at random. This can occur anywhere along the chromosome. The process of mutation, on the other hand, involves the switching of bits on a single chromosome at sites that are chosen at random. Each member of the next-generation chromosomal set has its own fitness function value established. This is done so that the optimal set of donor chromosomes can be chosen to replace the damaged or missing ones. In this study, we select the first 100 chromosomes, those with the lowest values for the fitness function (the size of the chromosomal population is 100). This is because they have the lowest fitness function values. This process of development and selection repeats itself over and over again for a number of generations, until the suggested fitness function achieves an asymptotic value and can no longer be improved upon.

C. Feature Selection Using Significance Scores

In high-dimensional spaces, the GA assesses subsets of features or SMART qualities to determine their value. Based on prior studies' findings, it's plausible to assume that these characteristics exhibit good group behaviour, which would manifest as distinct clusters in the feature space when considered collectively. On their own, any of these factors may or may not be particularly significant in establishing whether or not a drive has failed. Therefore, a straightforward mechanism is suggested in order to independently examine each feature that is chosen by the GA. This evaluation would involve calculating each feature's significance score, which would provide a rough measurement of the feature's contribution to the failure of disk drives.

D. Classification Using the Naive Bayes Classifier

The two-tier feature selection technique trains the NB classifier to discriminate between healthy and failing HDDs using GA-selected features and feature significance scores. Bayes' rule is applied to a generative classifier to categorize Xn using the NB classifier.

$$Py=cXn,\theta\alpha PXny=c,\theta P(y=c|\theta) \quad (8)$$

With class labels in hand, the NB classifier makes the assumption of feature independence under those conditions. This aids in reducing the amount of open-ended estimates involving unknown parameters. This is a reasonable assumption to make, considering the great majority of features in the SMART dataset [15] do not show any form of link with one another. The NB classifier is able to predict class labels for unknown vectors of the input SMART features by using the Bayes rule and computing the joint probability of the SMART attributes and the class labels. These forecasts make use of the binary success/failure categories of "Healthy" and "Failing." In order to predict a failing disk drive, the SMART data is not optimal for use with a discriminative classifier such as a support vector machine (SVM), which would merely transfer the feature vectors to the output labels. As a result, sensitive students may have trouble grasping the minority group's class composition.

IV. Experimental Setup and Results

As a result of the GA, the initial feature space has had its dimensionality reduced from 42 to 12. It does this by optimizing the fitness function using a population of 100 chromosomes that is passed down for a maximum of 80 generations. Table 1 presents the selected subset of features based on the results of the GA.

Table 1: The Genetic Algorithm has chosen these specific features

S. No.	SMART ID	Attribute Name
1	11	Power Cycle Count
2	177	Reported Uncorrected Errors
3	192	Temperature
4	196	Current Pending Sector Count
5	188	Uncorrectable Sector Count
6	5	Spin-up Time
7	6	Start/Stop Count
8	4	High Fly Writes
9	179	Seek Error Rate

10	163	Load/Unload Cycle Count
11	12	Spin Retry Count

The GA will select a subset of the traits, which may work well together but may include less-than-ideal features. It's possible that the presence of these less-than-ideal characteristics is not a significant factor in judging whether or not an HDD is failing. As a result, feature significance scores will be produced for each of the GA-selected features as part of the proposed two-tier feature selection approach.

The reported uncorrectable failures, spin-up time, and spin retry count are tracked by the SMART features with serial numbers 2, 6, and 11, respectively. The second tier of the proposed procedure for selecting features throws out these features. Table 2 presents the conclusive list of the nine characteristics that were chosen using the two-tiered feature selection approach that was proposed. The results presented in Table 3 provide conclusive evidence of the viability of the suggested two-tier system for feature selection. Table 2 shows the feature selections for the recommended two-tier strategy for training an NB classifier, which yields a 99.01 percent classification accuracy and a 0.24% false positive rate (FPR). Tabular data shows this. The NB classifier performs poorly when trained with numerous SMART attribute sets.

Table 2: Two-tier feature selection process features.

S. No.	SMART ID	Attribute Name
1	189	High Fly Writes
2	12	Power Cycle Count
3	194	Temperature
4	193	Load/Unload Cycle Count
5	198	Uncorrectable Sector Count
6	197	Current Pending Sector Count
7	4	Start/Stop Count
8	7	Seek Error Rate

The GA's feature selection is effective, as seen by the 92% average accuracy and 0.92 % FAR, however it does include some SMART parameters that are not major determinants in identifying a faulty disk drive. The differences between NB and a discriminative classifier like a support vector machine are laid out in Table 3. When trained using only the features chosen by the GA, the SVM is able to get an accuracy of 83.30 percent on average and a false alarm rate of 0.26%. When trained with the selected nine features, the proposed two-tier approach improves average accuracy by a little margin. On the other side, it shows that the TPR and the FPR have been falling. There is a decline in total patient ratio (TPR) to 44% and a slight increase in FPR (0.6%). According to Table 3, the NB classifier is superior to the SVM model for forecasting disk drive failure. The diagnostic performance of the HDD is improved when the SMART attributes are chosen using the proposed two-tier feature selection method, as shown by testing with two distinct types of classifiers.

As was said previously, the support vector machine (SVM) is a discriminative classifier, which means that it represents the direct link that exists between the feature vectors and the class labels. When dealing with SMART data, where there isn't always a clear-cut causal connection between the two variables, this strategy might not be the most effective one to use. As can be seen in Figure 6 and 7, both the SVM and NB classifiers exhibit positive ROC curves. The results of both classifiers are shown in this picture as well.

Table 3: Naive Bayes and Support Vector Machine classifier performance utilizing different feature selection strategies

Method	Feature Vector Dimensionality	True Positive Rate (%)	No. of Folds for Cross Validation	False Positive Rate (%)	No. of Test Iterations	Average Accuracy (%)
Proposed Two-Tier Method using SVM	9	44.	3	0.6	10	84.3
Proposed Two-Tier Method using Naive Baye	9	98.4	3	0.24	10	99.01
Naive Baye using GA only	12	72.0	3	0.92	10	92.0
SVM without Feature Selection	42	75.0	3	74.95	10	20.56
SVM with GA only	12	40.040	3	0.260	10	83.3
Naive Baye with No Feature Selection	42	55.255	3	1.031	10	86.98

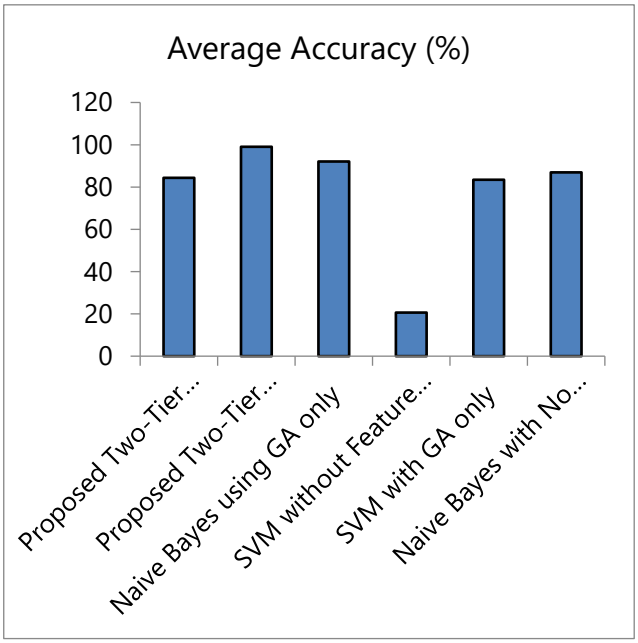

Fig 6: SVM and Naive Bayes Classifier average Accuracy

When trained on the nine features selected using the proposed two-tier feature selection strategy (as shown in Figure 5's ROC curves), the NB classifier achieves the highest TPR and FPR. This is demonstrated by the fact that the ROC curves for these two metrics are identical. The proposed method has a number of benefits that set it apart from other approaches that have already been utilized to identify malfunctioning hard disks. Using a two-stage feature selection approach, the suggested method identified the nine SMART features shown in Table 2 as the best predictors of a hard drive failure.

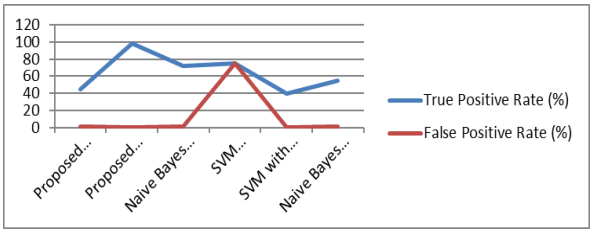

Fig 7: SVM and Naive Bayes Classifier curves with different feature selection strategies

The time needed to train the classifier is reduced as a result. This is accomplished. 1500 of the 2065 hard disks are functioning normally, while the remaining 565 have failed [1]. These are organized in three different folds, each of which has 678, 678, and 679 hard drives in total. Each fold has a total of 500 good drives, while the remaining drives have been damaged in some way. The proposed method successfully identified 185 of 188 failed drives, with a false positive rate of only 1.2%. An additional major improvement brought about by the suggested algorithm is its applicability to online HDD diagnostics. The values of an HDD's nine SMART properties, which are outlined in Table 2, can be fed into a trained instance of either a neural network or a support vector machine classifier. Thus, the classifier's output might be interpreted as "HDD in good health" or "HDD about to fail."

V. Conclusion

This article presents an innovative two-tier approach to identifying the most significant precursors to an HDD failure. The method was developed to raise the proportion of true positives. Data from 21 models ranging in size from 1.0TB to 8.0TB was collected over the period of nine months in a commercial datacentre, and the resulting set of 42 SMART properties was used as a starting point for this analysis. The list of 42 SMART attributes was used to select these precursors. The SMART qualities were analysed not just in combination but also on an individual basis using the proposed two-tiered technique. To begin, a GA was utilized to investigate the numerous feature subspaces in an effort to identify the SMART attribute combination that yielded the greatest results. The quality of the feature subset was determined by examining feature samples and determining how strongly they clustered for the two classes and how effectively the clusters were isolated from one another. The purpose of this was to evaluate the quality of the feature set. To get there, we compared each feature subset's value for intra-class compactness to that of inter-class separation and choose the one with the lowest value. The Euclidean distance was used to determine both the degree of compactness within a class as well as the degree of separation between any two classes. To personally analyze the qualities that were chosen by the GA, a new metric called the significance score was suggested in the second tier of the evaluation process. The significance score quantified the amount of statistical influence that a certain attribute had on the number of failed disks. The elimination of features that had statistical scores that were lower than a predetermined value. The proposed two-tier feature selection procedure resulted in the final list of features selected consisting of nine SMART properties as opposed to the initial 42 attributes, which led to a reduction in the amount of time required for the classifiers to complete their training. After that, a generative classifier known as the NB was trained using these nine attributes as input. The NB provided an FDR of 98.40%, in contrast to the discriminative classifier that the SVM provided, which was just 40.0%. In order to prevent the classifier from overfitting the majority class data, the HDD failure data was skewed by under-sampling the majority class of healthy HDD. This was done so that the classifier would not become overly sensitive to the majority class's data. The proposed method was successful in detecting 185 of the 188 failing drives, with an average of just 1.2 false alarms being generated. Finally, we can use this approach to place data blocks in Hadoop Distributed File System based on hard disk fitness.

References

[1] Coursey, G. Nath, S. Prabhu and S. Sengupta, "Remaining Useful Life Estimation of Hard Disk Drives using Bidirectional LSTM Networks," 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 2021, pp. 4832-4841, doi: 10.1109/BigData52589.2021.9671605.

[2] "A multi-instance LSTM network for failure detection of hard disk drives," 2020 IEEE 18th International Conference on Industrial Informatics (INDIN), Warwick, United Kingdom, 2020, pp. 709-712, doi: 10.1109/INDIN45582.2020.9442240.

[3] F. L. F. Pereira, I. Castro Chaves, J. P. P. Gomes and J. C. Machado, "Using Autoencoders for Anomaly Detection in Hard Disk Drives," 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 2020, pp. 1-7, doi: 10.1109/IJCNN48605.2020.9206689.

[4] Lee, C. & Cao, Yi & Ng, Kam K.H.. (2017). Big Data Analytics for Predictive Maintenance Strategies. 10.4018/978-1-5225-0956-1.ch004.

[5] G. Wang, Y. Wang and X. Sun, "Multi-Instance Deep Learning Based on Attention Mechanism for Failure Prediction of Unlabeled Hard Disk Drives," in IEEE Transactions on Instrumentation and Measurement, vol. 70, pp. 1-9, 2021, Art no. 3513509, doi: 10.1109/TIM.2021.3068180.

[6] J. Zeng, R. Ba, Q. Chen, L. Wu, H. Wang and Y. Xiong, "Prediction of Hard Drive Failures for Data Center Based on LightGBM," 2022 IEEE 9th International Conference on Cyber Security and Cloud Computing (CSCloud)/2022 IEEE 8th International Conference on Edge Computing and Scalable Cloud (EdgeCom), Xi'an, China, 2022, pp. 105-110, doi: 10.1109/CSCloud-EdgeCom54986.2022.00027.

[7] M. Simongyi and P. Chongstitvatana, "Machine Learning Methods for Abnormality Detection in Hard Disk Drive Assembly Process: Bi-LSTM, Wavelet-CNN and SVM," 2018 2nd European Conference on Electrical Engineering and Computer Science (EECS), Bern, Switzerland, 2018, pp. 392-399, doi: 10.1109/EECS.2018.00079.

[8] L. P. Queiroz et al., "A Fault Detection Method for Hard Disk Drives Based on Mixture of Gaussians and Nonparametric Statistics," in IEEE Transactions on Industrial Informatics, vol. 13, no. 2, pp. 542-550, April 2017, doi: 10.1109/TII.2016.2619180.

[9] Prafullata Auradkar et al., Performance tuning analysis of spatial operations on Spatial Hadoop cluster with SSD, Procedia Computer Science 167 (2020) 2253–2266

[10] Mukhtaj Khan, Zhengwen Huang, Maozhen Li, Gareth A. Taylor, Phillip M. Ashton, Mushtaq Khan, "Optimizing Hadoop Performance for Big Data Analytics in Smart Grid", Mathematical Problems in Engineering, vol. 2017, Article ID 2198262, 11 pages, 2017.

https://doi.org/10.1155/2017/2198262

[11] Ahmad, S.G., Liew, C.S., Munir, E.U., Ang, T.F. and Khan, S.U., (2016). A hybrid genetic algorithm for optimization of scheduling workflow applications in heterogeneous computing systems. Journal of Parallel and Distributed Computing, 87, pp.80-90.

[12] Archive.ics.uci.edu. (2019). UCI Machine Learning Repository: Bag of Words Data Set.

[13] [online] Available at: https://archive.ics.uci.edu/ml/datasets/bag+of+words [Accessed 9 Apr. 2019]

[14] Dai, W., Ibrahim, I. and Bassiouni, M., (2017), June. An improved replica placement policy for Hadoop Distributed File System running on Cloud platforms. In Cyber Security and Cloud Computing (CSCloud), 2017 IEEE 4th International Conference on (pp. 270-275). IEEE.

[15] Dharanipragada, J., Padala, S., Kammili, B. and Kumar, V., (2017). Tula: A disk latency aware balancing and block placement strategy for Hadoop. In Big Data (Big Data), 2017 IEEE International Conference on (pp. 2853-2858). IEEE.

[16] Docs.gluster.org. (2018). Home - Gluster Docs. [online] Available at: https://docs.gluster.org/en/latest/ [Accessed 21 Nov. 2018].

[17] Fahmy, M.M., Elghandour, I. and Nagi, M., (2016), December. CoS-HDFS: co-locating geo-distributed spatial data in hadoop distributed file system. In Proceedings of the 3rd IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (pp. 123-132). ACM.

[18] Kanemitsu, H., Hanada, M. and Nakazato, H., (2016). Clustering-based task scheduling in a large number of heterogeneous processors. IEEE Transactions on Parallel and Distributed Systems, 27(11), pp.3144-3157.

[19] Khaldi, D., Jouvelot, P. and Ancourt, C., (2015). Parallelizing with BDSC, a resource constrained scheduling algorithm for shared and distributed memory systems. Parallel Computing, 41, pp.66-89.

[20] Mrs. Ritika Dhabliya. (2020). Obstacle Detection and Text Recognition for Visually Impaired Person

Based on Raspberry Pi. International Journal of New Practices in Management and Engineering, 9(02), 01 - 07. https://doi.org/10.17762/ijnpme.v9i02.83

[21] Shukla, A., Almal, S., Gupta, A., Jain, R., Mishra, R., & Dhabliya, D. (2022). DL based system for on-board image classification in real time, applied to disaster mitigation. Paper presented at the PDGC 2022 - 2022 7th International Conference on Parallel, Distributed and Grid Computing, 663-668. doi:10.1109/PDGC56933.2022.10053139 Retrieved from www.scopus.com