Hierarchical Mahalanobis Distance Clustering Based Technique for Prognostics Applications Generating *Big Data*

R Krishnan* and S Jagannathan*
* Department of Electrical and Computer Engineering
Missouri University of Science and Technology
Rolla, Missouri- 65401
email: krm9c, sarangap@mst.edu

***Abstract*—In this paper, a Mahalanobis Distance (MD) based hierarchical clustering technique is proposed for prognostics in applications generating *Big Data*. This technique is shown to have the ability to overcome certain challenges concerning *Big Data* analysis. In this technique, Mahalanobis Taguchi Strategy (MTS) is utilized to generate MD values which are in turn organized into a tree. The hierarchical clustering approach is then applied to obtain an overall MD value which is trended over time for prediction. Simulation results are presented to demonstrate the efficiency of the proposed technique.**

I. Introduction

The next generation of digital manufacturing involves the fusion of complex physical machinery and networked sensors. Examples of such systems include health care, process control, manufacturing units, fleet of vehicles and so on [1]. Such complex systems are monitored for maintenance and in the process significant amount of data is generated.

For instance, a single cross-country flight in the United States generates an astonishing 240 terabytes of data [1] and this volume is ever growing. According to [2], the volume has gone from being in the range of Megabytes in the 70s to Exabytes today, where 1 Exabyte $= 2^{60}$ bytes.

This deluge of data (*Big Data*) is characterized by Volume (total size of the data present), Velocity (streaming data), Variety (range of data types and sources like numeric, pictures and so on), Veracity (quality of the data) and Value (the importance of the information present in the data) as shown in [3], these characteristics are collectively known as 5 V's of *Big Data*. By analyzing *Big Data*, valuable insight can be gathered about system performance, which can be used to increase its efficiency [4]. A multitude of challenges are to be addressed during this process, which include real-time analysis and its corresponding latency, memory management, system complexity, application variety or scalability and most importantly the 5 V's of *Big Data* [2].

Processing Big Data involves two major roadblocks, one is being able to load such data on the system memory for analysis. This road block was addressed in a seminal paper that introduced Google File System (GFS) in [5], which was later extended to Hadoop in [6]. Hadoop is a java based framework, that processes data in batches using a parallel computing environment like MapReduce [7]. Improving or developing technologies to this end is not our aim in this paper.

On the other hand, the other road block is the development of methodologies and techniques, that can efficiently analyze data and produce actionable insights. To this end, researchers have applied data mining methodologies [8] and multivariate statistical analysis techniques [9] for *Big Data* analysis. The use of these techniques for *Big Data* analysis is feasible but they are not suitable for prognostics applications where near real time processing for multi dimensional data is necessary. In [10], the high dimensionality of *Big Data* is tackled by using neural networks (NN), but this process is essentially complex and memory intensive and therefore cannot be used for near real-time analysis. All current data analyzing techniques could be applied on *Big Data* applications, but they are not tailored to attack the problem of *Big Data* prognostics where challenges like sensor reduction and placement, fault isolation and detection are to be addressed. Therefore to the best of our knowledge, no efficient technique is currently available for *Big Data* prognostics.

In [11], [12], Mahalanobis Taguchi System (MTS) is used as a tool for prognostics applications to detect bearing and centrifugal pump failures. The core structure of MTS [13] is based on a distance metric known as Mahalanobis Distance (MD) [14]. The MD has the ability to fuse data from multiple attributes into one single performance metric, which represents the distinction between two points in a multi-dimensional space. MD also has the ability to capture overall trends in the data. When MD deviates from the healthy scenario, an anomaly is said to be detected. By clustering MD values into different clusters, one can isolate the type of anomaly, that has occurred.

As a consequence, MD can be used to tackle the multi-dimensionality of *Big Data* near real-time and trending of the overall MD value can result in fault prognostics. Orthogonal arrays and signal to noise ratio analysis can be used for sensor reduction and placement. These characteristics of MTS show promise and indicate that hierarchical MTS can overcome *Big*

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.82

Data challenges. Also to the best of our knowledge, MTS has never been used for *Big Data*.

One of the reasons, MTS in its current form cannot be used for *Big Data* directly is because calculation of MD values involves correlating different attributes present in the data resulting in a complex correlation matrix. This matrix is very big and subsequently storage/handling of this matrix becomes a huge problem. By using a tree [15] like structure and combining MD values, the size of correlation matrix can be kept small and data organization can be improved. Consequentially, in this paper, a novel MD based hierarchical clustering technique for prognostics is proposed, this technique builds upon the principles of MTS to analyze *Big Data* for prognostics. The rest of the paper is organized as follows. Section II describes the proposed methodology, and Section III details the simulation results. Section IV provides conclusions of the paper.

II. Hierarchical MD Based Clustering

The process of *Big Data* prognostics is divided into four steps, which includes data collection, storage, feature extraction and finally analysis/prognostics. In this paper, data collection, storage and feature extraction are not targeted, therefore it is assumed that the *Big Data* set under test is divided into multiple segments for storage and each segment is assumed to be consisting of m observation and n attributes. It is also assumed that each segment in the dataset can be transferred to the system memory at least one at a time.

Let the data from a segment at time t be represented by $D(t)$, where $D(t) \in \mathbb{R}^{m\times n}$ is the data matrix : $m, n \in \mathbb{N}$, where $\mathbb{R}$ is the set of real numbers, $\mathbb{N}$ is the set of natural numbers, m is the total number of observations present and n is the total number of attributes present.

Every row in $D(t)$ is a data point and it is represented by a value known as Hierarchical Mahalanobis Distance (HMD). To calculate this value, a binary tree known as $Mdtree$ is used. The advantage of using a binary tree to cluster MD values is that the size of the correlation matrix can be fixed at 2×2.

Let $Mdtree = (L, N) : L, N \in \mathbb{N}$, where L is the depth of $Mdtree$ and $N = 2^{L+1} - 1$, where N is the total number of nodes in the tree.

$Mdtree$ is composed of α leaf nodes (nodes, that do not have a child node connected to it) and $(N - \alpha)$ internal nodes (any node, that has a child node connected to it) called I. Each leaf node in the $Mdtree$, represents an attribute.

Each Internal node I_η in $Mdtree, \forall\eta \in \{1, 2, 3, ..., (N - \alpha)\}$ has exactly 2 child nodes, $\{\gamma_1, \gamma_2\}$. I_η is composed of a Node Mahalanobis Distance (NMD) value, normal mean d, normal standard deviation s and correlation matrix C at that node. NMD value at I_η is calculated as,

$$NMD_\eta = \frac{1}{2} Z C^{-1} Z^T, \tag{1}$$

where C is the correlation matrix and Z is the normalized data matrix, which is given by

$$Z_{ij} = \frac{NMD_{ij} - \overline{d_j}}{s_j}, \tag{2}$$

where $j \in \{\gamma_1, \gamma_2\}$ and $i \in \{1, 2, 3, ..., m\}$, NMD_{ij} is the MD value at the j_{th} child node and i_{th} observation and Z_{ij} is the normalized data point at the j_{th} node and i_{th} observation. Mean ($\overline{d_j}$) is calculated by using eq: 3 and standard deviation s_j is calculated by eq: 4.

$$\overline{d_j} = \frac{\sum_{i=1}^{m} NMD_{ij}}{m} \tag{3}$$

$$s_j = \sqrt{\frac{\sum_{i=1}^{m} (D(t)_{ij} - \overline{d_j})^2}{m - 1}}, \tag{4}$$

where $D(t)_{ij}$ is the data point at the i_{th} observation and j_{th} node, C is given by eq: 5

$$C_{ij} = \frac{\sum_{p=1}^{n} (Z_{ip} * Z_{jp})}{n - 1}, \tag{5}$$

where C_{ij} is the member of the correlation matrix C at the i_{th} row and j_{th} column. C^{-1} is given by taking the inverse of C. An example $Mdtree$ is shown in Fig: 1.

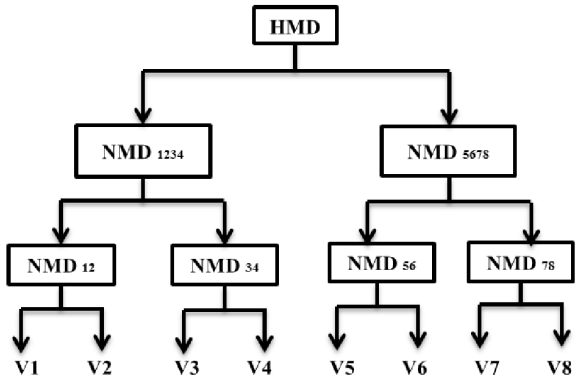


Fig. 1: Example $Mdtree$.

A. Methodology

The methodology of hierarchical MD based Clustering is divided into three stages.

1) Stage 1. Reference HMD Space: Let $D_N(t) \subset D(t)$ be the data sample pertaining to the normal region of operation for the system under test such that $D_N(t) \in \mathbb{R}^{p\times n} : p, n \in \mathbb{N}$ and t is the current time instant. $Mdtree$ is initialized first such that, n columns in data $D_N(t)$, are assigned to n leaf nodes in the tree. Each node in the tree is evaluated and in the process NMD, $\overline{d}$, s and C are calculated and stored at each node. HMD_i, $\forall i \in \{1, 2, 3, .., p\}$ is obtained and as a result reference HMD space is established.

2) Stage 2. Sensor Reduction and Placement: Once the reference HMD space has been established, optimization of attributes is done. For this, it is assumed that certain amount of data is available that belongs to a particular fault. An orthogonal array is first initialized [16], $OA \in \mathbb{R}^{2^n \times n}$, where 2^n is the total number of runs in the experiment. This array is based on Taguchi's design of experiments principles and it outlines 2^n experimental combination, that can be used

to determine which combination is sufficient to analyze a particular fault. Using eq: 1, HMD values are calculated for each row combination in OA. Using these HMD values, signal to noise ratio is calculated for the included attributes and the excluded attributes separately by

$$SN_\beta = -10\log\left[\frac{1}{m}\sum_{j=1}^{m}\frac{1}{HMD_j}\right],$$

for $HMD_\beta = \{HMD_\omega\}$, where $\omega \in \{1,2,3,4,...,m\}$ and $\beta \in \{1,2,3,4,...,2^n\}$. Depending on these two signal to noise ratios, gain is calculated for each row of OA, this gain represents the degree of closeness for the current experimental run from the row combination in which, all attributes are included in analysis, this is given by,

$$Gain = (SN_\beta)_{level_1} - (SN_\beta)_{level_2} \tag{6}$$

When gains for all the rows in OA are obtained then, the set of optimized attributes Γ is given by

$$\Gamma = \{\delta : \delta = \{1,2,3,...,2^n\}, Gain_\delta \geq 0\} \tag{7}$$

An example of all the calculated gains is shown in Fig: 2.

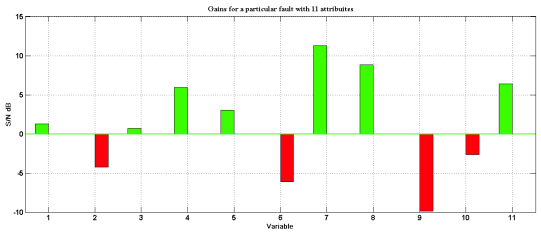


Fig. 2: Example signal to noise ratio's gain plot, green indicates that the gain is positive and red indicates that the gain is negative.

Optimized number of attributes are determined by selecting those attributes, that have a positive gain. Optimized number of attributes are unique for a particular fault. It must be noted that when *Big Data* is targeted, the total number of attributes are quite significant and using this analysis, unwanted and expensive sensors can be eliminated.

3) Stage 3. $Mdtree$ Initialization: $Mdtree$ is reinitialized depending on the optimized attributes set, corresponding to the fault. Using this tree HMD values are calculated for the data. A flow chart of this process is shown in Fig: 3. These HMD values are then clustered based on predefined fault cluster centroids enabling isolation and detection of fault.

B. Hierarchical Mahalanobis Distance (HMD) Based Fault Clustering

Let $C = \{c_1, c_2, c_3, ..., c_k\}$ be the set of all the centroid HMD values for each of the faults for the system under test, where k is the total number of faults. A particular HMD value is clustered based on the cluster centroid index (r), which is given by,

$$r = \{\underset{\forall i \in \{1,2,3...,k\}}{\arg\min} |HMD - c_i|\} \tag{8}$$

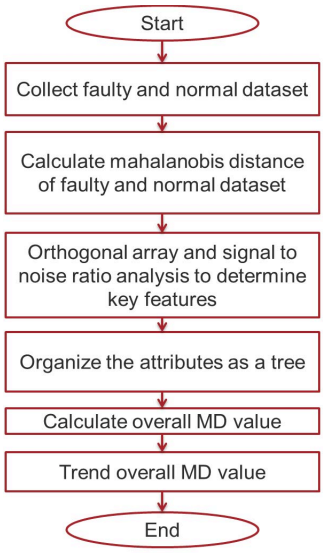


Fig. 3: Flow chart of Hierarchical MD based clustering.

For the purpose of clustering, centroids of the clusters are defined prior to clustering based on expert knowledge about the trends of a particular fault. In other words, data from healthy and faulty operation is utilized to construct these clusters. In this paper, the cluster centers represent all the members of the cluster and these compact clusters essentially follow a normal distribution. However any outlier can change the size of the cluster and therefore must be eliminated. In this work, any outliers are eliminated after computing the corresponding HMD values.

An outlier is defined as an observation point, which is very different from other observations in the dataset. Let $C = \{c_1, c_2, c_3, ..., c_k\}$ be the set of all the centroid HMD values for each of the faults for the system under test, where k is the total number of faults. Let $F = \{f_1 \cup f_2 \cup f_3 . \cup, ..., \cup f_k\}$, where f_k is the set of all the members of a particular fault, $\cup$ is the union operator, μ and ν are the mean and standard deviations of all the HMD values of the members of a fault cluster f_k. A HMD value is said to be an outlier, if the condition

$$|HMD - \mu| < \Lambda\nu, \tag{9}$$

where $\Lambda \in \mathbb{R}$ is the multiplicative factor, is satisfied. A cluster is assumed to follow a standard normal distribution and consequentially this factor is three.

C. Hierarchical Mahalanobis Distance (HMD) Based Prognostics

In this study, trending HMD values are used to determine the progression of a system from a normal region of operation to a faulty region of operation. The reference HMD space is a unit space and therefore, if the HMD is much greater than one, then the system is said to be moving away from normalcy and towards a fault.

In the next section, the aforementioned claims about the proposed technique are substantiated by simulation results.

III. Rolling Element Bearing Data set for CNC Machines

A. Design Of Experiments

This dataset is derived from a CNC machine test bed constructed at the Missouri S&T in 2009. The Taig micro mill CNC machine, that is used to generate this dataset houses two Ortadogu Rulaman Sanayai (ORS) 6023 brand rolling element bearing. Each bearing is attached with temperature and vibration sensors. Four types of conditions are introduced manually into the system and the data is collected.

1. *Normal Condition*: The machine is operated with new bearings installed.
2. *Cage Defect*: All the lubrication is removed from the bearings.
3. *Inner Race Defect*: A groove is cut into the inner race of the bearings
4. *Outer Race Defect*: A groove is cut into the outer race of the bearings

11 features are extracted from sensor data. These are (1) Cage Defect Frequencies in the x and the y direction (CD_x, CD_y). (2) Ball Pass Outer Race Frequencies in the x and the y direction $(BPFO_x, BPFO_y)$. (3) Ball Pass Inner Race Frequencies in the x and the y direction $(BPFI_x, BPFI_y)$. (4) Root Mean Square and Kurtosis of the vibration signal in the x and the y direction $(RMS_x, RMS_y, KURT_x, KURT_y)$. (5) Temperature $(temp)$. The resulting dataset is of the size 364 Gb, details about the feature extraction can be reviewed from [11].

B. Step 1: Sensor Reduction and Placement

The initial hierarchical Mahalanobis space is constructed using 11 features. $Mdtree$ is initialized and orthogonal array analysis is performed, the optimization results are obtained and they are tabulated in TABLE: I. It is observed that, there

TABLE I: Total number of optimized variables for each fault under evaluation

Type of Fault	Number of Optimized Attributes
Cage Fault	8
Inner Race Fault	6
Outer Race Fault	6

is a significant reduction in the total number of attributes after optimization. This is very favorable for *Big Data* prognostics as this procedure leads to significant cost savings and increased reliability. The result of this orthogonal array analysis is shown in TABLE: IV. A value of *yes* indicates that the attribute is marked as favorable for that particular fault.

1) Scalability of the sample: The reference space is established using varying sizes of data sample like 100%, 50%, 25%,12.5% and 5% of the available data. An accurate reference space is observed for varying sample sizes, indicating that established reference space does not depend upon the size of the sample.

C. Step 2: Initialization of $Mdtree$

After optimization, a $Mdtree$ is generated for each of the faults. An example tree as generated for cage fault is shown in Fig: 4.

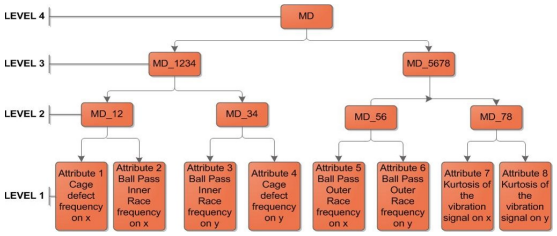


Fig. 4: Example $Mdtree$ for NO LUBE (Cage Fault) condition

After this step, fault clusters centroids are initialized to facilitate clustering of HMD values. These values are tabulated in TABLE: II. These centroids are dependent upon the predefined fault cluster.

TABLE II: Cluster centroids for each one of the faults

Type of Fault	Cluster Centroids Value (HMD)
Normal	1
Cage Fault	10
Inner Race Fault	30
Outer Race Fault	970

D. Step 3: Analysis

HMD values are calculated using $Mdtree$ and are clustered into various predefined clusters. These clusters are then analyzed for outliers.

1) Case 1: Outlier Detection: The proposed outlier detection technique is used to detect the outliers and these results are tabulated in TABLE: III. A multiplicative factor of three is used for analysis. It is observed that, all the outliers in the cluster are detected by our methodology. For experimentation purposes, outliers are introduced into the clusters and analysis is done. This is done because the data is too well behaved. Fault clusters after outlier detection are plotted in Fig: 6.

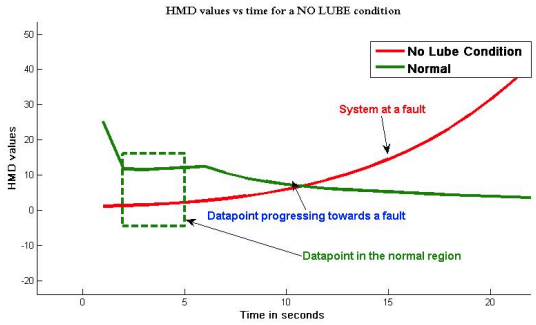
2) Case 2: HMD Trend: The trends of HMD values for various types of faults with respect to time are plotted in Fig: 5. It is seen from the plot that for the normal range of operation, the resulting HMD values are very small and almost close to one, where as for the various types of faults, the HMD's are

TABLE III: Outlier Detection Results

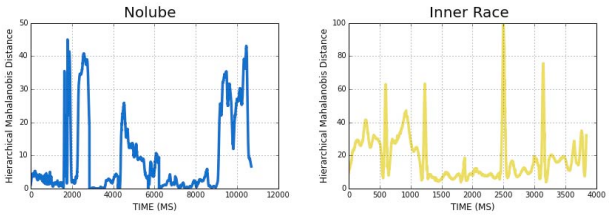
	Total elements	Outliers	Detection rate
1	12022	0	100%
2	11201	11201	100%
3	11497	3942	97%
4	12122	100	100%

TABLE IV: Orthogonal Array Results

Type of Fault	CD_x	CD_y	$BPFI_x$	$BPFI_y$	$BPFO_x$	$BPFO_y$	$Kurt_x$	$Kurt_y$	RMS_x	RMS_y	$Temp$
Cage Fault	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes	No
Inner Race Fault	No	Yes	No	No	Yes	No	No	No	Yes	Yes	No
Outer Race Fault	Yes	No	Yes	Yes	No	Yes	No	No	Yes	Yes	No

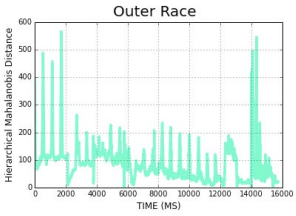


(a) HMD values progressing towards a fault.

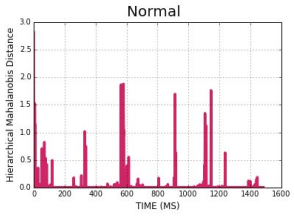


(b) No Lube v/s Time

(c) Inner Race v/s Time



(d) Outer Race v/s Time



(e) Normal v/s Time

Fig. 5: Trending of HMD values for various faults.

much greater than the HMD's in the normal region, which validates our reference space. It is also seen from the plots that different faults have different range of values for HMD and this indicates that the proposed technique can be used to isolate these faults. It must be observed here that, due to the nature of the proposed technique, the final HMD values are one dimensional. Due to this property, it is noted that the clusters are linearly separable and boundary conditions/thresholds can be clearly defined depending on the trends, which assists in fault analysis.

3) Case 3: Prognostics: Clear progression of a system going from a normal region of operation towards a fault is seen in Fig: 5a. Fig: 5a corresponds to the cage fault condition and this plot demonstrates the prognostic ability of the technique. Isolation of a fault is feasible if a maximum and a minimum

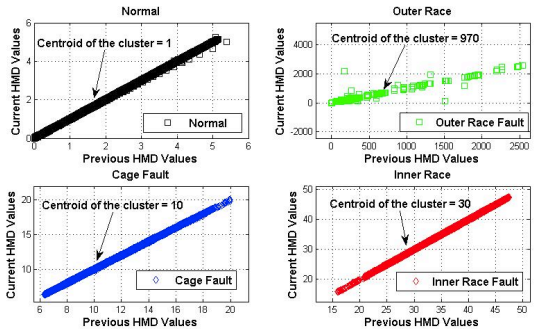


Fig. 6: HMD Clusters

for a particular fault is defined, this is easy in this case because every cluster has a maximum and a minimum value. The maximum value for a single cluster can be used as a threshold for the system response. In other words, if HMD values are greater than a maximum value of a certain cluster, that means the system is moving away from that particular fault and similarly, if it is greater than a certain minimum, then the system is moving towards that particular fault.

4) Case 4: Cluster Validity Metrics: To evaluate the performance of our technique, it is necessary to evaluate the formation of fault clusters by using some standard validation metrics. Various cluster validity metrics are present in the literature and these are explained in [17]. Out of the many present in the literature, the most relevant metrics for this study are compactness, separation and dunn validity Index. Let k be the total number of clusters defined by the proposed technique, $F = \bigcup_{i=1}^{k} \{f_i\}$, where f_i represents a fault cluster, $i \in 1, 2, 3, ..., k$ and $C = c_1, c_2, c_3, ..., c_k$ is the set of all the cluster centroids.

a) Compactness (CP): This metric measures the closeness of the cluster. Let CP_i be the compactness of fault cluster given by

$$\overline{CP_i} = \frac{1}{|f_i|} \sum_{x_j \in f_i} ||x_j - c_i||, \tag{10}$$

where $|.|$ is the total number of elements present in the set and $||.||$ is the euclidean norm. Compactness $\overline{CP}$, that is the average compactness for all the clusters is found by eq: 11.

$$\overline{CP} = \frac{1}{k} \sum_{i=1}^{k} \overline{CP_i} \tag{11}$$

A lower value of compactness is better as it indicate the existence of compact clusters.

TABLE V: Cluster Validity Metrics

Validity Index	Average value
Compactness	0.001
Separation	75.7
Dunn Validity Index	5.003

b) Separation (SP): This metric quantifies the distinction between the different clusters. The value is calculated by using eq: 12

$$\overline{SP} = \frac{2}{k^2 - k} \sum_{i=1}^{k} \sum_{j=i+1}^{k} ||c_i - c_j|| \tag{12}$$

where $i \in 1, ..., k$, $j \in 1, ..., k$. The more the value is greater than zero, larger is the separation of clusters.

c) Dunn Validity Index (DVI): This metric quantifies both the degree of separation and the degree of compactness in one single metric.DVI is given by eq: 13

$$DVI = \frac{\min\limits_{0<p\neq q<k} \{ \min\limits_{\forall c_i \in C_p, \forall c_j \in C_q} \{||c_i - c_j||\}\}}{\max\limits_{0<p\leq k} \max\limits_{\forall x_i, x_j \in f_p} \{||x_i - x_j||\}} \tag{13}$$

A larger value of DVI indicate compact and well separated clusters.

Its seen from TABLE: V, that the value of compactness for this dataset comes out to be very low and the value of separation is also large. These values indicate an acceptable performance of the proposed technique in defining clusters.

IV. Conclusion

In this paper, a hierarchical MD based technique is introduced for *Big Data* prognostics. The advantages of the proposed technique include the ability to handle the high dimensionality of *Big Data*, organization of the attributes resulting in better memory management, generic approach and the ability to perform sensor reduction and placement. The proposed technique can perform prognostics including fault isolation and detection. The ability to perform outlier detection in *Big Data* was also observed during analysis.

The performance of this technique is validated in this paper with rolling element bearing dataset. It is observed both from the approach and from the case study that expert domain knowledge is required to make informative decisions in the preprocessing stage and certain amount of data is required for establishing reference space. This technique might not be suitable for a completely unknown data due to the lack of expert knowledge for determining thresholds. We aim to improve this technique to handle various types of data and to tackle other challenges of *Big Data* that were not addressed in this paper.

Acknowledgment

This research was supported in part by NSF I/UCRC award IIP 1134721 and Intelligent Systems Center.

References

[1] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, A. H. Byers, and M. G. Institute, "Big data: The next frontier for innovation, competition, and productivity," 2011.

[2] H. Hu, Y. Wen, T. Chua, and X. Li, "Towards scalable systems for big data analytics: A technology tutorial," 2014.

[3] P. Zikopoulos, C. Eaton *et al.*, *Understanding big data: Analytics for enterprise class hadoop and streaming data*. McGraw-Hill Osborne Media, 2011.

[4] P. Baraldi, F. Cadini, F. Mangili, and E. Zio, "Model-based and data-driven prognostics under different available information," *Probabilistic Engineering Mechanics*, vol. 32, pp. 66–79, 2013.

[5] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The google file system," in *ACM SIGOPS operating systems review*, vol. 37, no. 5. ACM, 2003, pp. 29–43.

[6] A. Abouzeid, K. Bajda-Pawlikowski, D. Abadi, A. Silberschatz, and A. Rasin, "Hadoopdb: an architectural hybrid of mapreduce and dbms technologies for analytical workloads," *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 922–933, 2009.

[7] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[8] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 26, no. 1, pp. 97–107, 2014.

[9] R. A. Johnson, D. W. Wichern *et al.*, *Applied multivariate statistical analysis*. Prentice hall Englewood Cliffs, NJ, 1992, vol. 4.

[10] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[11] A. Soylemezoglu, S. Jagannathan, and C. Saygin, "Mahalanobis-taguchi system as a multi-sensor based decision making prognostics tool for centrifugal pump failures," *Reliability, IEEE Transactions on*, vol. 60, no. 4, pp. 864–878, 2011.

[12] ——, "Mahalanobis taguchi system (mts) as a prognostics tool for rolling element bearing failures," *Journal of Manufacturing Science and Engineering*, vol. 132, no. 5, p. 051014, 2010.

[13] G. Taguchi and R. Jugulum, *The Mahalanobis-Taguchi strategy: A pattern technology system*. John Wiley & Sons, 2002.

[14] P. C. Mahalanobis, "On the generalized distance in statistics," *Proceedings of the National Institute of Sciences (Calcutta)*, vol. 2, pp. 49–55, 1936.

[15] W. H. Day and H. Edelsbrunner, "Efficient algorithms for agglomerative hierarchical clustering methods," *Journal of classification*, vol. 1, no. 1, pp. 7–24, 1984.

[16] A. S. Hedayat, N. J. A. Sloane, and J. Stufken, *Orthogonal arrays: theory and applications*. Springer Science & Business Media, 2012.

[17] A. Fahad, N. Alshatri, Z. Tari, A. ALAmri, A. Y Zomaya, I. Khalil, S. Foufou, and A. Bouras, "A survey of clustering algorithms for big data: Taxonomy & empirical analysis," 2014.