www.s2is.org
International Journal on Smart Sensing and Intelligent Systems
ISSN 1178-5608

# Ranking of Sensitive Positions using Statistical and Correlational Analysis

**[1]Nishchal K. Verma, [1]Rahul K. Sevakula, [1]Sonal Dixit, [2]Al Salour**

[1]Department of Electrical Engineering, Indian Institute of Technology Kanpur, India
[2]Boeing Company, St. Louis, MO, USA

E-Mail : nishchal@iitk.ac.in, srahulk@iitk.ac.in, dsonal@iitk.ac.in, al.salour@boeing.com

***Submitted: June 10, 2013 Accepted: Aug 08, 2013 Published: Sep. 05, 2013***

***Abstract – Condition Based Monitoring of a machine refers to the checking of various parameters and signatures of the machine and then predicting the machine's current health status. The slight changes in the machine operating condition are carefully analyzed to know if the machine is having any fault. For monitoring the changes, we acquire acoustic data i.e. generated while machine is running. In order to collect acoustic data, a number of sensors are placed near various positions of the machine surface. Acquiring data from large number of sensor positions is not economically viable. It would always be preferable to have a monitoring system that acquires data quickly and efficiently, without compromising on the robustness of the system. Therefore there is a need to locate some special positions on the machine, termed as "sensitive positions", which are expected to exhibit the fault characteristics in a much better way than others. This paper presents a novel method for ranking sensitive positions for a machine based on statistical parameters analysis. While the final list of required number of sensitive positions is generated, the cross-correlation amongst the positions is also taken into consideration to avoid redundancy. Furthermore, a standalone application for implementing the same has been developed on Android platform. The proposed scheme and application can be used for variety of other applications that work on similar principle of acquiring data from multiple sensors.***

**Index terms – Condition Based Monitoring, Data Acquisition, Sensitive Positions, Statistical Parameters, Android, Sensors**

# I. INTRODUCTION

Finding an optimal machine maintenance strategy is one of the major concerns of large scale industries. Condition Based Maintenance is one such effective maintenance strategy [1] which ensures reduced breakdown time, reduced maintenance cost and better reliability of system. Condition based monitoring does not inhibit the failure process; instead it helps in early detection of faults, thus preventing catastrophic equipment failure.

To determine the current condition of running systems, multiple machine signatures i.e. acoustic, vibrational, temperature, pressure signatures are acquired and analyzed [2][3]. Whenever machine degrades, these signatures get affected. Different sensors i.e. acoustic sensor, vibrational sensor, etc. have been designed to collect different signatures of machine [4]. Quality data acquisition is an important aspect of condition based monitoring [5]. In this research work, we use the machine's acoustic signature to distinguish between the different machine conditions. To capture acoustics, microphone is placed near the surface of machine. While recording machine sound, position of microphone also matters. This is because some positions have clearer signal, as compared to others.

As illustrated in previous research work, on/around the surface of machine, there exist some positions that give better signal than others, termed as sensitive positions. Verma et-al's paper [6] describes a statistical approach for finding the sensitive position on air compressor. In this approach, whole compressor was divided into four sides. Then acoustic signatures were taken from number of points across each part. A total of 24 points were marked, 6 on each side of the compressor. 4 sensitive positions (one for each surface) were found based on standard deviation, variance and *rms* values of the acquired signal. Resultant positions were further analyzed for their measure of dissimilarity with other positions using cross correlation.

Despite good results, this approach has a limitation of choosing only one sensitive position on each surface. The approach is very specific to their application and cannot be generalized for all machines. Also, the results are not flexible enough, as we get set of 4 sensitive positions for a machine. It was also noticed that cross correlation amongst positions should be used while determining the sensitive position, instead of using it only for verification.

To address all above mentioned issues, we propose here an enhanced and generalized algorithm, where machine is not divided into parts, instead considered as one unit. All the positions are then ranked on the basis of individual statistical parameters simultaneously [7]. To avoid redundancy of acquired signal, we perform cross correlation analysis to get uncorrelated sensitive positions.

Along with the algorithm, we have also made an Android [8] based application for ranking sensitive positions using the same algorithm. Using this application on smartphone [9-11] gives one the flexibility of easily finding sensitive positions of a machine. The acoustics of the machine is recorded using the smartphone’s inbuilt microphone.

The biggest advantage of using Android platform is that it allows both data acquisition and analysis on the same platform. Application programming interfaces (APIs) allow third party applications to have good integration with the phone’s operating system. These features of Android as an operating system inspired us to make a standalone, robust application for sensitive position analysis.

The rest of the paper is organized as follows. Section II describes in detail the algorithm used for ranking. Section III explains the implementation details of Android. Section IV summarizes the results obtained and Section V concludes the paper.

## II. THE ALGORITHM

The proposed ranking algorithm consists of four main steps. Figure 1 shows the flowchart of the proposed algorithm.

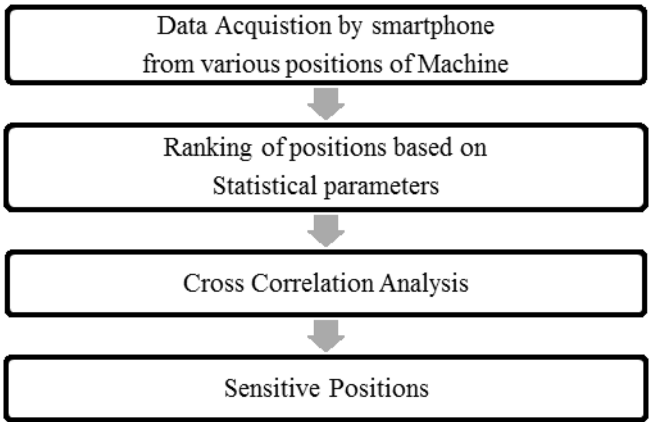


Figure1. Flowchart of the proposed algorithm

### a. Data Acquisition

In order to collect dynamic information about machine, data acquisition (DAQ) is to be done with utmost care. The sound emanated from the machine under running condition is recorded by the smartphone and saved in Waveform Audio File (WAV) format. This audio format is one of the few audio formats that store acoustic data without compression and complex coding; making it easier for us to get the sampled signal data [12]. WAV format stores sampled acoustic data in 16 bit 2's compliment PCM format. Customized code was written to capture acoustic data at 44.1 kHz (sampling frequency of smartphone that is guaranteed to work on all devices) [13] and store it in .wav format. In this way, the smartphone with its microphone act as a DAQ module.

### b. Pre-processing of the Signals

The recorded samples inevitably contain noise from the surroundings, which result in a smaller signal to noise ratio. By pre-processing, the effect of noise and outliers can be reduced [14]. The following pre-processing steps were used for the model:-

1.) Down Sampling: Raw data was collected at the rate of 44.1 kHz. Thorough experimentation, which was aimed to establish the relation between each fault and its dominant frequency band, showed that information regarding fault was found in frequencies below 12 kHz [15]. Frequency components above 12 KHz are not very informative; hence they can be neglected during analysis. Therefore, we down sample the data by a factor of 2. This will help in reduction of data for computation in future stages, without much compromise on the information content.

2.) Clipping: Down sampled signal is further reduced in size by the process of clipping. For performing clipping, the 5 second signal is divided into 9 parts on time scale (0-1, 0.5-1.5, 1-2 & so on). Then standard deviation is calculated for every part of the signal. The part (1 second) of the signal having least standard deviation is selected, as this part is expected to be more stable than others.

3.) Filtering: Previous research [15] showed that most of the useful information for the air compressor in acoustic data is contained below frequencies of 12 KHz. In order to keep only relevant information and to avoid high frequency noise, a low pass Butterworth filter [16] with a cut off frequency of 12 kHz was used.

4.) Smoothing: A simple averaging filter was used for smoothening of data; thus reducing the effect of outliers.

### c. Ranking of positions based on Statistical Parameters

We have used the following notations – ‘N’ denotes the total number of sensor positions and ‘$n_s$’ represents the number of sensitive positions required. The pre-processed signal which consist of 1 second of sampled data i.e. 22050 samples, following statistical parameters as summarized in Table I, were calculated.

1.) Peak - It gives the maximum magnitude of the given sampled set.

2.) Standard Deviation - It is a measure of average deviation from mean.

3.) Root Mean Square (RMS) - It is defined as the square root of the arithmetic mean of the squared signal.

4.) Absolute Mean- Since the processed data consists of both positive and negative values, mean of the absolute processed data was taken into consideration.

Table I. Statistical Parameters

| | |
|---|---|
| Peak | $\max(X_i)$ |
| Standard Deviation | $\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(X_i-\bar{X}\right)^2}$ |
| Root Mean Square(RMS) | $\sqrt{\frac{1}{N}\sum_{i=1}^{n}X_i^2}$ |
| Absolute Mean | $\frac{1}{N}\sum_{i=1}^{n}\lvert X_i\rvert$ |

According to the magnitude of each parameter, sensitive positions were ranked separately for each parameter in descending order. Thus each sensor position has 4 ranks namely $r_{i1}$, $r_{i2}$, $r_{i3}$ *and* $r_{i4}$, where position with higher parameter values gets better rank (i.e. lower rank value). If position ‘*A*’ is having the highest parameter value then it will get rank = 1. After getting the 4 ranks for each position, using equation (1), accumulated rank, *r_acc* is found.

$$r\_acc_i = w_1 r_{i1} + w_2 r_{i2} + w_3 r_{i3} + w_4 r_{i4} \tag{1}$$

where, $r\_acc_i$ is the accumulated rank of ith sensor position and $w_1$, $w_2$, $w_3$, $w_4$ are the weights for ranks given by individual statistical parameters, namely peak, standard deviation, root mean square and absolute mean respectively. For simplicity, here all weights have been taken equal to 1. $r_{i1}$ is the rank of position ‘i’ with respect to statistical parameter 1. The accumulated ranks are then arranged in ascending order and the sensor positions are given ranks accordingly.

d. Cross Correlation Analysis

Cross Correlation is a measure of the level of similarity between two time domain signals. Cross correlation is given by equation (2).

$$r_{XY} = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)S_X S_Y} \qquad (2)$$

$S_x$ and $S_y$ are the standard deviations of time domain signals '$x$' and '$y$' respectively. $X_i$ *and* $Y_i$ refers to the $i^{th}$ sample of signals '$x$' and '$y$' respectively. $r_{XY}$ is the cross correlation coefficient between signals '$x$' and '$y$', $\bar{X}, \bar{Y}$ are the mean values of '$x$' and '$y$' respectively and '$n$' is the number of time samples of the signal.

The strength of linear relationship between the two time domain signals is maximum when cross correlation value is ±1 and the same diminishes when value approaches 0. To analyze whether the two sensor positions are giving redundant information, a threshold value needs to be set for the cross correlation coefficient. If the value of coefficient between the two sensor positions is greater than the threshold, the two positions can be considered as sufficiently similar. This means, only one of these positions need to be present in the final list of sensitive positions as keeping both of them increases redundancy; hence reduced amount of information.

1.) Threshold value for Correlation

To get cross correlation coefficients, cross correlation operation is performed on the pre-processed data corresponding to different positions. The cross correlation values for each pair of sensor positions are sorted in descending order. If '$N$' is the number of sensor positions and '$n_s$' is the required number of sensitive positions, we take average of the $(n_s\text{-}1)^{th}$ highest correlation value for each sensor position. This value is then set as the threshold for cross correlation analysis. The reason we take average of $(n_s\text{-}1)^{th}$ correlation values is that in a list having $n_s$ positions, each position will at most be checked with $(n_s\text{-}1)$ sensor positions for cross correlation analysis. As the $(n_s\text{-}1)^{th}$ correlation value will generally not be same for all sensor positions, we take their average as the threshold value.

2.) Elimination based on Correlation Analysis

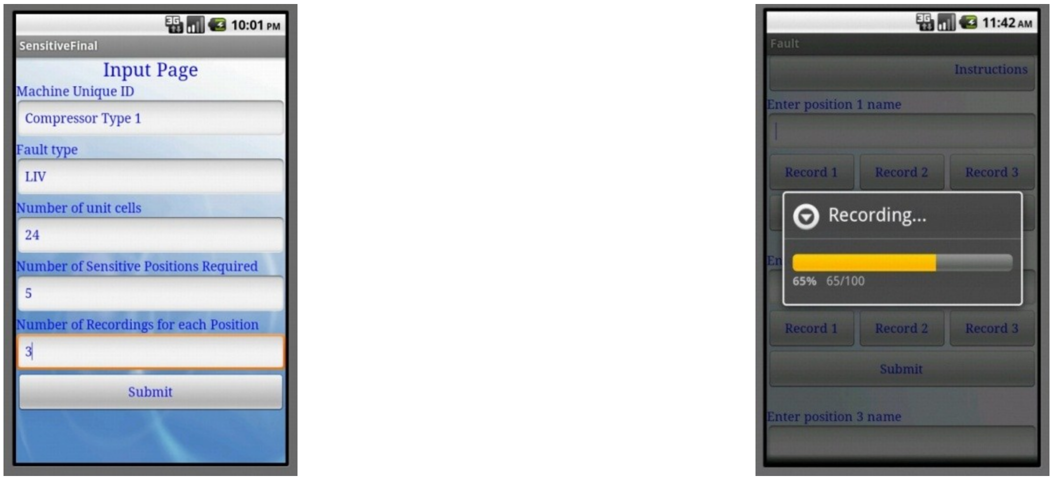
Before finding the cross correlation coefficients, we ranked the sensor positions as per their accumulated rank, *r_acc*. As we are in need of $n_s$ sensitive positions, we start with $n_s$ highest ranked sensitive positions, as chosen from the list. Starting from the highest ranked sensor position, the cross correlation coefficient of higher ranked position with lower ranked positions is compared with the threshold that has been set in the previous stage. In case, the cross correlation coefficient with any of the lower ranked position is higher than the threshold (which means correlation is high), then the lower ranked sensor position is removed from the list and the next best ranked sensor position (which was initially not present in the selected list) is added to the bottom of this selected list. The same procedure is continued with other selected sensor positions as well, until a stage, where we have a list of sensitive positions that are not only sensitive, but also sufficiently uncorrelated. It must be noted that the algorithm is designed for $n_s << N$.

## III. IMPLEMENTATION

We started with the recording of acoustic signals from different compressor positions. As we had first implemented our proposed scheme on MATLAB, the sensitive positions were found offline on a desktop PC using MATLAB. After getting satisfactory results in MATLAB, the same algorithm was then implemented on Android platform. A new app is developed on Android SDK platform [8], in plug with Eclipse IDE [17] using java. The application is made in a user friendly manner, such that one can take data from different positions and find the sensitive position immediately. Developed application consists of activity pages, namely Input page, Data Acquisition Page, Camera Activity page and Result page. In first activity page, user is asked to give Machine's Unique ID, Fault/Health Status detail, number of Sensor Positions ($N$), required number of Sensitive Positions ($n_s$) and number of recordings per position ($n_r$). All this information is used for back end implementation of the algorithm.

If the type of fault is not specified, the application assumes the machine to be healthy. Here, we need to take machine status as an input, because the list of sensitive positions need not be same for different machine conditions. Further pages are then dynamically generated based on these inputs. In case, analysis has already been done for the given machine ID and fault detail, the application shows the stored results and also gives an option to overwrite the stored results. The Data

Acquisition page of the application interacts with the user to take ‘$n_r$’ recordings, each of 5 seconds duration, for each sensor position. Here, we give a facility for having multiple recordings per sensor. This is because multiple recordings improve accuracy of the analysis, in the sense that if one recording was improper, other recordings give us opportunity to correct it. The recordings are temporarily saved as ‘.wav’ format in the SD card of phone. Immediately after each recording, the signal is preprocessed and parameters are extracted and stored in a matrix for future use. These 4 parameters were then averaged over all recordings for that position. The recording whose standard deviation was closest to the mean standard deviation of ‘$n_r$’ recordings was chosen while calculating cross-correlation. A horizontal progress bar was shown to make the user aware of the progress of recording and parameter calculation. The application is made complete by putting various checks, for e.g. in case all the recordings corresponding to any position are not taken, a dialog box containing an alert message is displayed.



Input Page
Machine Unique ID: Compressor Type 1
Fault type: LIV
Number of unit cells: 24
Number of Sensitive Positions Required: 5
Number of Recordings for each Position: 3

Recording...
65% 65/100

(a) (b)

Figure2. a) Input page of Application, b) Data Acquisition page showing the recording process

After data acquisition, two threads run in parallel. The back end of the application was implemented in a separate thread to produce the required number of sensitive positions. The UI (User Interface) thread kept a track of the progress via a horizontal progress bar. In Java Coding, most of the operations were performed using the built-in packages. For calculating correlation, “commons-math-2.2” and “commons-io-2.2” packages of Apache Commons Mathematics Library [18] were used. Once the list of sensitive positions is generated, the Camera Activity page of the application is activated which then gives user, the option to take photographs for all sensitive positions using the smart phone's built-in camera. These images are stored at a resolution of (640 x 480) so that they can be used for future reference. Finally, the Result page displays the sensitive positions along with their images. The user is then given an option to save these results with reference to the given unique machine id and class. All these details including the images are then stored in a directory named after Machine ID and class in SD card. The developed application was tested on a single stage reciprocating air compressor present in lab for the following 3 conditions.

1) Condition of machine
   - Healthy condition
   - Condition of LIV fault (Leakage of inlet valve)
   - Condition of LOV fault( Leakage of outlet valve)
   - Condition of NRV fault (Non return valve)
2) Specifications of compressor
   - Air Pressure Range: 0-500 lb/m$^2$, 0-35 Kg/cm$^2$
   - Induction Motor: 5 HP, 415V, 5Amp, 3ph, 50 Hz, 1440 rpm
   - Pressure Switch: Type: PR-15, Range: 100-213 PSI, 7-15 Kg/cm$^2$
3) Specifications of smart phone
   - Android OS, v 2.3.5(Gingerbread)
   - 830 MHz ARM V6 CPU, 290 MB user available RAM
4) Packages used
   - Android Developer Tools (ADT) plug-in
   - JAVA IDE
   - Software development Kit

The number of sensor positions 'N' to be checked was 24 and number of required sensitive positions '$n_s$' were 5. The number of recordings for each position '$n_r$' was set as 1.

## IV. RESULTS

Tables II, III, IV and V summarize the results of the application (using proposed algorithm) on the acoustic data obtained from a compressor running in healthy condition. Table II lists the average statistical parameters obtained from the recordings of 24 sensor positions. Ranks of sensor positions on the basis of individual parameters and overall rank on the basis of accumulated ranks is shown in Table III. For correlation analysis, the threshold correlation value for the data set was 0.0397. For different running condition, i.e. where dataset is different, the threshold for correlation value will be different.

Table II. Statistical Parameters of recordings at 24 positions of machine in Healthy condition

| Peak | S.D.* | RMS | Abs. Mean | S.P.* |
|---|---|---|---|---|
| 0.195471 | 0.028951 | 0.029244 | 0.021786 | 1 |
| 0.24949 | 0.03457 | 0.034808 | 0.024899 | 2 |
| 0.3405 | 0.051307 | 0.051471 | 0.035416 | 3 |
| 0.170986 | 0.028335 | 0.028625 | 0.021616 | 4 |
| 0.233201 | 0.037472 | 0.037711 | 0.027713 | 5 |
| 0.239653 | 0.032244 | 0.03251 | 0.023638 | 6 |
| 0.168627 | 0.033543 | 0.03379 | 0.025902 | 7 |
| 0.234604 | 0.039236 | 0.039452 | 0.029941 | 8 |
| 0.212889 | 0.044389 | 0.044587 | 0.034318 | 9 |
| 0.190743 | 0.037957 | 0.03818 | 0.029335 | 10 |
| 0.432649 | 0.050174 | 0.05034 | 0.035324 | 11 |
| 0.357053 | 0.060715 | 0.060858 | 0.047175 | 12 |
| 0.572658 | 0.08467 | 0.084765 | 0.055211 | 13 |
| 0.339079 | 0.064253 | 0.06438 | 0.045899 | 14 |
| 0.288145 | 0.050495 | 0.050668 | 0.036968 | 15 |
| 0.310192 | 0.05466 | 0.054811 | 0.040454 | 16 |
| 0.479369 | 0.064468 | 0.064593 | 0.047687 | 17 |
| 0.557491 | 0.080883 | 0.08099 | 0.059829 | 18 |
| 0.151912 | 0.031483 | 0.031761 | 0.024448 | 19 |
| 0.211592 | 0.039465 | 0.039672 | 0.030213 | 20 |
| 0.183559 | 0.035397 | 0.035623 | 0.026986 | 21 |
| 0.245633 | 0.041721 | 0.041916 | 0.031692 | 22 |
| 0.531311 | 0.085096 | 0.085194 | 0.059245 | 23 |
| 0.507022 | 0.077057 | 0.07717 | 0.057432 | 24 |

**S.D. =Standard Deviation, S.P. =Sensor Position*

The correlation values between the top five sensitive positions have been shown in Table IV. The correlation between the pair (23, 24) was higher than the threshold value. So, position 24 was removed from the list of top five sensitive and was pushed to the bottom of the list of rankings. Position 12, which was ranked 6th, was appended into the list. Again correlation coefficients of position 12 with 23, 13, 18 and 17 were checked. As some of the values were above the threshold, position 12 was eliminated and next position 14 was included. The same process was followed for position 14 and position 16. After three iterations, top five sensitive positions were obtained for healthy condition as shown in Table V.

Table III. Ranking of Sensor positions on the basis of individual parameters and Overall rank of positions for Healthy condition

| S.P. | Peak Rank | S.D. Rank | RMS Rank | Mean Rank | Accumulated Rank | Overall Rank |
|---|---|---|---|---|---|---|
| 23 | 1 | 1 | 3 | 1 | 6 | 1 |
| 13 | 4 | 3 | 1 | 3 | 11 | 2 |
| 18 | 2 | 2 | 6 | 2 | 12 | 3 |
| 24 | 6 | 5 | 2 | 5 | 18 | 4 |
| 17 | 3 | 4 | 11 | 4 | 22 | 5 |
| 12 | 7 | 7 | 4 | 7 | 25 | 6 |
| 14 | 5 | 6 | 13 | 6 | 30 | 7 |
| 16 | 9 | 9 | 5 | 9 | 32 | 8 |
| 3 | 8 | 8 | 9 | 8 | 33 | 9 |
| 11 | 10 | 10 | 10 | 10 | 40 | 10 |
| 15 | 11 | 11 | 7 | 11 | 40 | 11 |
| 22 | 13 | 13 | 8 | 13 | 47 | 12 |
| 9 | 12 | 12 | 12 | 12 | 48 | 13 |
| 8 | 16 | 15 | 14 | 15 | 60 | 14 |
| 20 | 14 | 14 | 19 | 14 | 61 | 15 |
| 5 | 15 | 16 | 17 | 16 | 64 | 16 |
| 10 | 17 | 17 | 15 | 17 | 66 | 17 |
| 2 | 18 | 19 | 16 | 19 | 72 | 18 |
| 21 | 19 | 18 | 18 | 18 | 73 | 19 |
| 6 | 20 | 20 | 20 | 20 | 80 | 20 |
| 7 | 22 | 21 | 21 | 21 | 85 | 21 |
| 1 | 21 | 23 | 24 | 23 | 91 | 22 |
| 19 | 24 | 22 | 23 | 22 | 91 | 23 |
| 4 | 23 | 24 | 22 | 24 | 93 | 24 |

Table IV. Cross Correlation values between the most sensitive positions of Healthy condition

| Sensitive Position Pair | Cross Correlation Value |
|---|---|
| 23,13 | 0.005 |
| 23,18 | 0.0034 |
| 23,24 | **0.0579** |
| 23,17 | 0.011 |
| 13,18 | **0.0464** |
| 13,24 | 0.0138 |
| 13,17 | 0.0204 |
| 18,24 | 0.0107 |
| 18,17 | **0.0441** |

Table V. Final List of Sensitive positions for Healthy condition after Cross Correlation Analysis

| Position | Rank |
|---|---|
| **23** | 1 |
| **13** | 2 |
| **17** | 3 |
| **16** | 4 |
| **3** | 5 |

Similarly sensitive positions were found for three different faulty states of machine, namely leakage inlet valve, leakage outlet valve and non-return valve. Intermediate and final results for these different machine states are given in tables VI, VII, VII, IX, X, XI and XII.

Table VI. Statistical Parameters of recordings at 24 positions of machine in LIV fault condition

| Peak | S.D. | RMS | Abs. Mean | S.P. |
|---|---|---|---|---|
| 0.964459 | 0.180309 | 0.180305 | 0.134497 | 1 |
| 0.965811 | 0.160952 | 0.160948 | 0.118243 | 2 |
| 0.956048 | 0.178355 | 0.178352 | 0.1312 | 3 |
| 1.009667 | 0.176803 | 0.1768 | 0.128518 | 4 |
| 0.969515 | 0.177792 | 0.177788 | 0.132986 | 5 |
| 0.9806 | 0.225675 | 0.225669 | 0.16841 | 6 |
| 0.972471 | 0.181703 | 0.181712 | 0.128302 | 7 |
| 0.806044 | 0.143106 | 0.143103 | 0.106392 | 8 |
| 1.027242 | 0.158947 | 0.158945 | 0.113442 | 9 |
| 1.052496 | 0.167691 | 0.167688 | 0.115512 | 10 |
| 1.072517 | 0.209061 | 0.209056 | 0.142702 | 11 |
| 1.081686 | 0.243102 | 0.243099 | 0.181956 | 12 |
| 1.010265 | 0.267962 | 0.267959 | 0.202682 | 13 |
| 0.986315 | 0.235633 | 0.235628 | 0.178452 | 14 |
| 0.951667 | 0.227285 | 0.22728 | 0.173746 | 15 |
| 1.0722 | 0.243017 | 0.243012 | 0.184641 | 16 |
| 1.08281 | 0.249441 | 0.249437 | 0.183423 | 17 |
| 1.092866 | 0.29436 | 0.294354 | 0.218832 | 18 |
| 0.812905 | 0.153079 | 0.153076 | 0.115477 | 19 |
| 0.939806 | 0.164821 | 0.164818 | 0.122663 | 20 |
| 0.931551 | 0.1742 | 0.174196 | 0.129338 | 21 |
| 0.876565 | 0.178105 | 0.178101 | 0.130858 | 22 |
| 1.059117 | 0.267934 | 0.267928 | 0.198385 | 23 |
| 1.087051 | 0.30306 | 0.303054 | 0.222437 | 24 |

Table VII. Ranking of Sensor positions on the basis of individual parameters and Overall rank of positions for LIV fault condition

| S.P. | Peak Rank | S.D. Rank | RMS Rank | Mean Rank | Accumulated Rank | Overall Rank |
|---|---|---|---|---|---|---|
| 24 | 1 | 1 | 1 | 1 | 4 | 1 |
| 18 | 3 | 2 | 2 | 2 | 9 | 2 |
| 13 | 6 | 5 | 5 | 5 | 21 | 3 |
| 17 | 5 | 4 | 9 | 4 | 22 | 4 |
| 23 | 2 | 3 | 16 | 3 | 24 | 5 |
| 12 | 9 | 7 | 4 | 7 | 27 | 6 |
| 16 | 4 | 6 | 17 | 6 | 33 | 7 |
| 14 | 10 | 10 | 3 | 10 | 33 | 8 |
| 11 | 8 | 8 | 14 | 8 | 38 | 9 |
| 6 | 11 | 11 | 7 | 11 | 40 | 10 |
| 15 | 12 | 9 | 10 | 9 | 40 | 11 |
| 1 | 7 | 13 | 11 | 13 | 44 | 12 |
| 7 | 13 | 12 | 8 | 12 | 45 | 13 |
| 3 | 16 | 16 | 6 | 16 | 54 | 14 |
| 5 | 15 | 15 | 12 | 15 | 57 | 15 |
| 4 | 14 | 14 | 20 | 14 | 62 | 16 |
| 10 | 17 | 17 | 13 | 17 | 64 | 17 |
| 22 | 20 | 18 | 19 | 18 | 75 | 18 |
| 21 | 18 | 19 | 21 | 19 | 77 | 19 |
| 9 | 22 | 22 | 15 | 22 | 81 | 20 |
| 2 | 19 | 20 | 22 | 20 | 81 | 21 |
| 20 | 21 | 21 | 18 | 21 | 81 | 22 |
| 19 | 23 | 23 | 24 | 23 | 93 | 23 |
| 8 | 24 | 24 | 23 | 24 | 95 | 24 |

Table VIII. Cross Correlation values between the most sensitive positions for LIV fault condition

| Sensitive Position Pair | Cross Correlation Value |
|---|---|
| 24,18 | 0.015 |
| 24,13 | **0.0386** |
| 24,17 | 0.0011 |
| 24,23 | **0.0266** |
| 18,13 | 0.0086 |
| 18,17 | **0.105** |
| 18,23 | **0.0367** |
| 13,17 | 0.000098 |
| 13,23 | 0.02 |
| 17,23 | 0.0071 |

Table IX. Most sensitive positions for LIV fault condition after cross correlation analysis

| Position | Rank |
|---|---|
| **24** | 1 |
| **18** | 2 |
| **12** | 3 |
| **16** | 4 |
| **14** | 5 |

Table X. Statistical Parameters of recordings at 24 positions of machine in LOV fault condition

| Peak | S.D. | RMS | Abs Mean | Rank |
|---|---|---|---|---|
| 0.802021 | 0.156298 | 0.156294 | 0.117239 | 1 |
| 0.730464 | 0.142008 | 0.142005 | 0.105347 | 2 |
| 0.840429 | 0.165144 | 0.16514 | 0.121719 | 3 |
| 0.881766 | 0.174713 | 0.17471 | 0.128046 | 4 |
| 0.79116 | 0.162575 | 0.162571 | 0.118771 | 5 |
| 1.010467 | 0.224028 | 0.224024 | 0.158489 | 6 |
| 0.737771 | 0.135261 | 0.135258 | 0.103575 | 7 |
| 0.711826 | 0.1259 | 0.125898 | 0.097018 | 8 |
| 0.788389 | 0.137868 | 0.137866 | 0.102723 | 9 |
| 1.022107 | 0.158347 | 0.158344 | 0.107802 | 10 |
| 1.078217 | 0.218837 | 0.218834 | 0.149285 | 11 |
| 1.03332 | 0.213121 | 0.213117 | 0.157969 | 12 |
| 1.029932 | 0.266817 | 0.266811 | 0.192073 | 13 |
| 0.989579 | 0.2401 | 0.240095 | 0.176166 | 14 |
| 1.007143 | 0.229369 | 0.229363 | 0.170029 | 15 |
| 0.976663 | 0.209837 | 0.209832 | 0.15582 | 16 |
| 1.035034 | 0.244165 | 0.24416 | 0.179743 | 17 |
| 1.091654 | 0.282499 | 0.282493 | 0.210158 | 18 |
| 0.840023 | 0.148144 | 0.148141 | 0.111531 | 19 |
| 0.947018 | 0.178481 | 0.178477 | 0.132141 | 20 |
| 1.008825 | 0.176483 | 0.176479 | 0.127539 | 21 |
| 1.019212 | 0.19528 | 0.195277 | 0.140951 | 22 |
| 1.082171 | 0.313017 | 0.313015 | 0.230739 | 23 |
| 1.073333 | 0.324571 | 0.324564 | 0.238889 | 24 |

Table XI. Most sensitive positions for LOV fault condition after cross correlation analysis.

| Position | Rank |
|---|---|
| **24** | 1 |
| **23** | 2 |
| **18** | 3 |
| **13** | 4 |
| **17** | 5 |

Table XII. Statistical Parameters of recordings at 24 positions of machine in NRV fault condition

| Peak | S.D. | RMS | Abs. Mean | S.P. |
|---|---|---|---|---|
| 0.705533 | 0.138173 | 0.138171 | 0.106236 | 1 |
| 0.994971 | 0.164023 | 0.164019 | 0.118951 | 2 |
| 0.753097 | 0.140743 | 0.14074 | 0.106446 | 3 |
| 0.897972 | 0.147559 | 0.147556 | 0.108636 | 4 |
| 0.957707 | 0.145816 | 0.145813 | 0.105455 | 5 |
| 1.028799 | 0.206412 | 0.20641 | 0.146688 | 6 |
| 1.031301 | 0.21121 | 0.211208 | 0.151427 | 7 |
| 1.099757 | 0.319655 | 0.319654 | 0.231116 | 8 |
| 1.163391 | 0.458642 | 0.458665 | 0.342123 | 9 |
| 0.994536 | 0.184574 | 0.18457 | 0.136683 | 10 |
| 1.080552 | 0.302142 | 0.302135 | 0.222698 | 11 |
| 1.065981 | 0.314352 | 0.314345 | 0.233768 | 12 |
| 1.032428 | 0.229095 | 0.22909 | 0.160954 | 13 |
| 1.000035 | 0.213741 | 0.213736 | 0.154198 | 14 |
| 0.948636 | 0.201874 | 0.20187 | 0.148835 | 15 |
| 0.973575 | 0.214422 | 0.214418 | 0.160759 | 16 |
| 1.042311 | 0.241112 | 0.241107 | 0.182423 | 17 |
| 0.745659 | 0.126432 | 0.126429 | 0.096537 | 18 |
| 0.713899 | 0.125766 | 0.125763 | 0.095577 | 19 |
| 0.964164 | 0.18347 | 0.183466 | 0.140126 | 20 |
| 1.012533 | 0.180428 | 0.180425 | 0.131419 | 21 |
| 0.905024 | 0.130248 | 0.130246 | 0.096027 | 22 |
| 0.746855 | 0.137686 | 0.137684 | 0.106257 | 23 |
| 0.73182 | 0.136782 | 0.136779 | 0.104889 | 24 |

Table XIII. Most sensitive positions for NRV fault condition after correlation analysis

| Position | Rank |
|---|---|
| **9** | 1 |
| **8** | 2 |
| **12** | 3 |
| **13** | 4 |
| **7** | 5 |

## V. CONCLUSION

In this paper we have proposed an original scheme for finding required number of sensitive positions with ranking, based on the statistical parameters of signals acquired from various positions. The sensitive positions found are not only sensitive, but also uncorrelated. They were also found to give clearer and louder recordings as determined by mechanics. The scheme was implemented as an application for Android based smartphones. For experimentation, real time recordings of reciprocating air compressor in four different running conditions were used. Though the experimentation and validation has been done on reciprocating type air compressors, we feel that the proposed algorithm can be used for a variety of applications that involve data acquisition from various sensors. For example: in hospitals while treatment, patients are monitored by collecting data from their bodies, e.g. EEG, ECG signals etc. It would be of great help, if we are able to find sensitive positions on patients' body, from where if data collected, can probably give better results. Further scope for improvement could include varied weights for ranks given by statistical parameters while calculating accumulated rank. Here we have taken all weights equal to one. Future work could also include validation of the algorithm on how well the system performs w.r.t classification accuracy in recognizing different machine conditions, while taking recordings from the sensitive positions found.

## ACKNOWLEDGEMENT

We would like to express our sincere gratitude towards Boeing for their funding support of this research work through research grant (BOEING/EE/20100220), at IIT Kanpur.

## REFERENCES

[1] Ken Fischer, "ENM 590 Case Studies in Engineering Management condition Based Maintenance Plus Return on Investment Analysis," US Army RDECOM-TARDEC Mile Rd Warren, MI USA, 12 Jan. 2011.

[2] Jardine, Andrew KS, Daming Lin, and Dragan Banjevic "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical systems and signal processing*, vol 20, no. 6, pp. 1483-1510, Oct 2006.

[3] S. Nandi, H. A. Toliyat, and Xiaodong Li "Condition monitoring and fault diagnosis of electrical motors-a review," *IEEE Trans. Energy Convers.*, vol 20, no. 4, pp. 719-729, 2005.

[4] Lee, Jay, Ramzi Abujamra, Andrew KS Jardine, Daming Lin, and Dragan Banjevic. "An integrated platform for diagnostics, prognostics and maintenance optimization," in Proc. of the Intelligent Maintenance Systems, pp. 15-27, 2004.

[5] N.V. Kirianaki, S.Y. Yurish, N.O. Shpak, and V.P. Deynega, Data Acquisition and Signal Processing for Smart Sensors, Wiley, Chichester, West Sussex, England, 2002.

[6] N.K. Verma, K. Jaganatham, A. Bhairat, T. Shukla, and A. Salour, "Finding sensitive sensor positions under faulty condition of reciprocating air compressors," *IEEE 2011 Recent Advances in Intelligent Computational Systems (RAICS)*, pp. 242-246, 22-24 Sept. 2011.

[7] N. K. Verma, P. Kumar, R.K. Sevakula, S. Dixit, A. Salour, "Ranking of sensitive positions based on statistical parameters and cross correlation analysis" in *Sixth IEEE International Conference Sensing Technology (ICST),* pp. 815-821, Dec 2012.

[8] Android Developers Guide, Available: http://developer.android.com/sdk/installing/index.html [Accessed: Jan 2012]

[9] R. Want, "When Cell Phones Become Computers", *IEEE Pervasive Computing*, vol. 8, no.2 , pp. 2-5, April 2009.

[10] Joshua J. Romero, "Smartphones: The Pocketable PC," IEEE Spectrum, Available: http://spectrum.ieee.org/telecom/wireless/smartphones-the-pocketable-pc [Accessed: Jan 2012]

[11] Jupiter research limited, "Next generation Smartphones players opportunities & forecasts 2008-2013," Feb 2009.

[12] Good and Best practices for making Audio Recordings, Available: http://archivehistory.jeksite.org/chapters/chapter6.htm [Accessed: July, 2012]

[13] Android Developers, Available: http://developer.android.com/reference/android/media/AudioRecord.html [Accessed: August, 2012]

[14] D. P. Ballou, and G. K. Tayi "Enhancing data quality in data warehouse environments," *Communications of ACM*, vol. 42, no. 1, pp. 73-78, Jan 1999.

[15] L. Sriram, "Audio based condition monitoring and fault diagnosis of Internal Combustion Engines," Master's thesis, Department of Electrical Engineering, IIT Kanpur, Kanpur India, 2008

[16] W. K. Chen, The Circuits and Filters Handbook, 3rd ed., Florida: CRC Press, Boca Raton, 1995.

[17] ECLIPSE JavaTM Developers Guide to Eclipse, Available: http://www.jdg2e.com/contents.html. [Accessed: August, 2012]

[18] Commons Math Developers, Apache Commons Math, Release 2.2 Available: http://commons.apache.org/math. [Accessed: August, 2012]