

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# An IoT Based Cloud Deployment Framework for Effective Classification of Machine Conditions

*Ganga Dhandapani and V. Ramachandran*

## Abstract

Cloud services are proposed for real-time data acquisition, data classification, data processing and decision making, which are highly interconnected services for effective condition monitoring of electrical machines. The proposed Software as a Service, Storage as a Service and Platform as a Service layers address the challenges of data storage and scalability while making analysis on the cluster of machines in an Industrial Environment. An experimental setup consisting of two DC motors coupled to AC Generator operating at different locations is considered to evolve the proposed model for effective integrated monitoring and decision making. This cloud-based vibration monitoring model provides services for data acquisition from the IoT devices mounted on the shafts of the DC motors, data storage to store the enormous amount of acquired signal data from multiple sensors, data classification of vibration signals for effective statistical analysis to estimate adaptive cluster of thresholds and appropriate decision-making services on demand over the Internet to utilize the reliable service of the machines in a persistent way. The computational engine will do inherent statistical analysis of the vibration signals to estimate the cluster of thresholds adaptive to various operating conditions. The services have been deployed without any limitation in a cloud environment and the industrial applications can share information using the deployed services from anywhere on demand basis. The deployed cloud service for the enhanced statistical classification algorithm eliminates the false identification of failures, which not only increase the availability of machines for intended operations but also reduce the maintenance cost. The resulting threshold values are compared with that of the vibration analysis carried out on the machine beds locally using myRIO for data acquisition in LabVIEW and the proposed model ensures the integrity in appropriate decision making with assured scalability.

**Keywords:** condition monitoring, electrical machines, data acquisition, classification, web services, Google Cloud, scalability

## 1. Introduction

The electrical machines find wide and crucial applications in various industries and power plants. Condition monitoring of electrical machines is extremely

significant for making the industrial processes more efficient with reduced downtime. Condition monitoring is heading as a real-time task, which requires maximum accuracy and embraces a gradual paradigm shift from legacy systems to modern Internet of Things (IoT) enabled systems at every level namely data acquisition, data processing, data integration and decision making. The uptime and efficiency of the plant operations shall be maximized through proper condition monitoring diagnostics and as well as preventive or predictive maintenance. The current condition monitoring systems make effective decisions using the knowledge repository, which is populated using various algorithms by the way of observing and storing the defective and unusual behaviour details of the machines. The major challenge faced by many industries is not only inadequate storage space but also the scalability when many machines inside the plant or operating at remote locations are to be monitored online and enormous amount of data have to be acquired from the machines for the interpretation of their behaviour at dynamic or abnormal operating conditions.

In the late 90's, very few online condition monitoring applications came into existence with the primary motive to collect vibration signals from various machines operating at different locations, but the analysis has been made considering each local operating environment to make effective decisions. This methodology leads to better preventive maintenance, but predictive maintenance is still a challenge. During those initial stages of online condition monitoring, the accessibility of such applications was through personal computers and laptops via World Wide Web. The scalability is the major issue in the current on-line condition monitoring applications. Over the period, due to the advancements in networking technologies, higher data rates for communication have paved a path to expansion in the field of cloud services. Especially, the industries which are facing the challenges such as inadequate storage space for data and scalability, the cloud environment will provide appropriate solution to those issues.

The cloud based model must be designed to handle operations by various industries without any hurdle to exchange data due to heterogeneous nature. The cloud environment provides inherent dynamic scalability for the operations of electrical machines at different locations at different operating conditions. Cloud computing does not require global standard architectures, and it does not necessarily need a standard, open, general purpose protocol. Furthermore, cloud computing supports interfaces that are syntactically simple, semantically restricted and of high-level. The cloud environment provides an added value of being able to share and compare the local machine condition data with other similar machines across the plant, or with other machines at multiple plants wherever they are located.

The growth of data analysis methods such as statistical, signal processing and machine learning techniques has moved condition monitoring of electrical machines towards the regime of predictive maintenance with the application of predictive analytics. Integrating the predictive maintenance techniques with the IoT enabled technologies will enable the industries to avoid unnecessary equipment replacement and improve process safety, availability and efficiency. Predictive maintenance adopted in industries employs predictive analytics to detect the problems well ahead to the occurrence of failures using which the corrective measures are planned. Prediction avoids unexpected process failures and prolongs the life of the system. In condition monitoring, though threshold estimation has more significance, it has not yet been given due consideration and thousands of false alarms are generated in dynamic operating conditions due to adoption of default threshold levels. The precise and faster short-term forecasting of machine's physical signals predict the probability of failures and intensity of deterioration during abnormal conditions and provide performance optimization under normal or dynamic

operating conditions. Condition monitoring stands effective only when the process of extracting information from the data becomes faster with more details. In this chapter, a new scalable and reliable model has been investigated to perform online condition monitoring of multiple machines in real-time industrial conditions and to perform predictive maintenance to enhance the process coordination and fault tolerance in industrial automation.

The statistical classification based vibration analysis algorithm has been developed as a Web service and deployed as a cloud service to demonstrate real-time condition monitoring of electrical machines. The proposed cloud based condition monitoring system collects the vibration data of machines from various locations and processes the same in the cloud by comparing the data of one machine with the data of other similar machines for reliable and effective decision making. These features of interfaces are underlying factors for rapid adoption of cloud computing services in the condition monitoring applications.

## **2. State of the art**

More research works regarding condition monitoring and predictive analysis are carried out for accurate assessment and prediction of machine conditions in real-time. The prediction models designed for monitoring real-time operation of electrical machines need to be robust and online in order to make accurate and faster data prediction.

It is hard to fulfill the practical requirement of application specific scientific approaches for the industries while performing real time data analysis and condition assessment towards preventive maintenance as the machines are operating at different environmental conditions. Diego Galar et al. [1] have cited that single valued thresholds provided by the manufactures are not suitable for fault identification under non-stationary operations, environmental changes and aging. The vibration severity characterized by ISO 10816 has been specified as a static threshold suitable for new machines and said to provide incorrect reference for machines in use. Instead, the authors provide dynamic thresholds adaptive to operational conditions as a better solution for SMART maintenance.

Continuous monitoring and measuring of machine parameters such as vibration, temperature, etc., with and without external disturbances will lead to make proper decisions for effective maintenance and thereby prolong the useful life and reliable operation of electrical machines. Recently, smart Internet of Things technologies are evolving for effective condition monitoring of electrical machines. IoT enables online monitoring of the machine as it runs and data have been acquired by an embedded device or a gateway and transmitted to a server for analysis and maintenance scheduling. The practical challenges faced by maintenance engineers are the introduction of new technologies for the enhancement of plant productivity, methods of data acquisition and analysis, inconsistent outcomes and shortage of resources. R. Kirubashankar et al. [2] has explained about Internet based automation architecture for the control of the devices and equipments of process plants for optimal control and reduced unplanned downtime. A Web based architecture has been proposed for the control of devices remotely over Ethernet with Programmable Logic Controller (PLC) and Supervisory Control and Data Acquisition (SCADA) system networked in Virtual Private Network (VPN). Larry Combs [3] has stated that the functionality and reliability of conventional SCADA gets enhanced if it is hosted as Software as a Service in cloud platform. IoT enabled condition monitoring will identify potential problems using sensors and able to take necessary preventive measures before any issues occur and hence prevent damages and reduce



maintenance costs. The changes in vibration, temperature etc., have been tracked by sensors and any issues such as misalignment, imbalance etc., shall be detected and accordingly service maintenance is scheduled automatically ahead of time to prevent failures thus avoiding unplanned downtime.

Omid Givvehchi et al. [4] had designed a general cloud based architectural model that allows automation functions in industries to be offered as services from a dynamic infrastructure. The authors have envisaged the importance of cloud solution for the control and field levels of automation. The physical devices at these levels are integrated by encapsulating the services and functions inside the delivery standards of cloud. Cloud computing is seen as solution to provide platform for integration of growing information technologies such as Internet of Things, Service Oriented Architectures and mobile computing. Omid Givvehchi and Jasperneite [5] have delivered the Virtual PLC as 'Control as a Service' through Microsoft Azure cloud environment. Omid Givvehchi et al. [6] have illustrated about the development of Virtual PLCs on the Virtual Machines of the private cloud created using VMware's vCloud suite.

Cloud technology can be applied in two ways for automation of industrial processes. They are collaborative application development and real time publishing of data to the cloud server for remote monitoring and control. Though real time publishing is adopted for remote monitoring, collaborative applications also has equal significance. This is because of the option available to decide the factors of design, customization, updates and changes in the system before deployment. Also collaboration allows multiple consumers to monitor the process data simultaneously. This is highly required in places where the systems are inherently distributed such as irrigation systems, wind farms, cell towers, agriculture etc. The SCADA system maintained as a central monitoring system in local network, when taken to cloud server with collaborative software tools can aid in easy information exchange at multiple locations. The real time publishing can be made so easy in SCADA using SaaS (Software as a Service) in cloud. The software running the application will be on cloud server enabling easy and secured data publishing and data request. This also sends the data to multiple clients such as iPad, smart phones or other networks. The SCADA built in cloud with software tools providing collaboration leads to the creation of new business model using direct and shared access of multiple processes to a control expert. Further, on the data received from industrial systems, analysis can be made in the server side and feedback can be provided [7].

In condition monitoring, the comparison of the data acquired with that of baseline standards is a widely adopted strategy [8]. NI in its artefact of fleet wide monitoring emphasizes the importance of continuous and automated data collection from industrial assets in order to realize meticulous comparative results so that real-time maintenance decisions are improved significantly. Such kind of maintenance strategy could successfully be achieved only with IoT based condition monitoring of industrial assets executed in cloud platform [9]. The cloud can be of public, private or hybrid nature. Fran Dougherty, CTO of the Worldwide Incubation Enterprise and Partner Group of Microsoft had appreciated the use of private and public clouds by industries for innovation, scalability and business growth in the special report composed by Jim Montague [10]. However, hybrid cloud was considered to be the best option by him, as industries can choose the type of analysis dynamically as per the requirements. Advantech in its white paper [11] has discussed on the importance of the implementation of cloud-based SCADA system using Industrial IoT and points out that the adoption of cloud offers pervasive analytics and decisions additionally irrespective of the hardware used and thus making Industry 4.0 effective. Steve Lacey [12] while discussing the ground realities of condition monitoring in industries asserted the need for skilled technicians

for predictive maintenance. The author perceived cloud based condition monitoring to expand the monitoring horizons and assure direct connectivity with maintenance specialists whose availability has always been a challenge and also hinted on the execution of cloud based condition monitoring in Schaeffler due to various benefits. New analysis techniques have been implemented when unknown signal patterns are observed at the user end.

The plant-wide condition monitoring of rotating electrical machines have extensively been discussed by Mallikarjun Kande et al. [13]. The existing machine condition monitoring and industrial automation techniques have been reviewed and the application of artificial intelligence for machinery diagnostics has been perceived as the future scope. While discussing about on-equipment and on premise integration methods, the need for on-cloud monitoring using IoT gateway has been substantiated to meet the requirements of advanced diagnostics and data platforms for enhanced computation. The condition monitoring system and Distributed Control System are integrated over the cloud for continuous monitoring of the equipment with high update rates from the sensors and for effective diagnosis. The efficient integration of various data acquisition and other devices in real time demands lightweight and uniform communication standards.

A comprehensive investigation has been made [14] that shifts the focus from the monitoring of specific machine components for fault prognosis to an approach scanning the overall system execution in an integrated manner to deliver desired performance for the application in an optimum manner. The authors address the challenge in the determination of absolute vibration thresholds adaptive to the machine operating conditions for reliable condition monitoring. A statistical classification based signal decomposition algorithm has been proposed for segmented vibration signal analysis as a measure of improving the precision in condition monitoring of electrical machines.

In this chapter, IoT based cloud services for real-time condition monitoring of electrical machines are proposed. It is focused towards estimation of vibration thresholds adaptive to the machine condition, which persuades to realization of incipient and critical abnormal conditions fully. Considering the immense raise towards the importance of predictive maintenance applications and connected IP based data acquisition devices, a generalized cloud framework is proposed to provide services for effective condition monitoring diagnostics and to maintain a knowledge repository for effective decision making with respect to maintenance scheduling. The main objective of this chapter is to explore real-time implementation of IoT enabled cloud services to formulate pre-emptive, strategic and operational decisions. The proposed IoT based model for vibration analytics of electrical machines addresses the challenges of data storage and scalability. A Web Application Framework has been developed by introducing cloud services for real-time data acquisition, data classification, data processing and decision making for effective condition monitoring.

The statistical classification based signal decomposition algorithm discussed in [14, 15] identifies the denser vibrating levels of machine under dynamic operating conditions and enumerates cluster of thresholds adaptive to the operating conditions for quick and accurate prediction of abnormalities. This algorithm is integrated with IoT based model through LabVIEW client application to enable real-time condition monitoring of machines located anywhere whose data are acquired by sensors and transmitted to cloud storage. The cloud services which have been developed for data acquisition and processing are tested locally before deploying in the Django Web Framework and implemented in Google Cloud Platform. The results of the classification algorithm, i.e., the adaptive threshold class clusters pertaining to each machine are used to create contextual vibration references for

making efficient and quick decisions in the condition assessment of machines of same type exposed to similar operating conditions.

### 3. Cloud services for effective condition monitoring

An effective cloud-based model is proposed to estimate the cluster of thresholds adaptive to various operating conditions of the electrical machines, to eliminate the identification of false failures or alarms, and to make decisions for effective maintenance scheduling. To make its implementation more general and scalable for real-time analysis, cloud services are introduced in different layers in accordance with Model-View-Template (MVT) pattern of Django Web Framework (<https://media.readthedocs.org>) and deployed in Google Cloud Platform. The “Models” represent the structure and manipulation of data, “Views” encapsulate the processes both at server and client ends and “Templates” present the rendered information to the end user. The basic building blocks of the proposed cloud-based model are to analyse the various dimensions and metrics such as, density of oscillations between classified amplitude levels, maximum number of oscillations, oscillations with similar and dissimilar amplitudes and the adaptive thresholds etc.

Cloud services are deployed for real-time data acquisition, data classification, data processing and decision making, which are highly interconnected services for effective condition monitoring of electrical machines. The proposed Software as a Service, Storage as a Service and Platform as a Service layers address the challenges of data storage and scalability while making analysis on the cluster of machines in an Industrial Environment. An experimental setup consisting of two DC motors each coupled to AC Generator operating at different locations is considered to evolve the proposed model for effective integrated monitoring and decision making. The threshold values estimated using cloud services are compared with that of the vibration analysis carried out on the machine beds locally using myRIO for data acquisition in LabVIEW ensures the integrity of the cloud-based model with assured scalability. Though security and big data processing overheads are encountered in the connected enterprises, employing cloud computing has been widely embraced by industries for its collaborative nature, optimized performance, better diagnostics, higher productivity and sustainability. IoT and cloud-based processing have been adopted for condition monitoring of multiple machines operating at different locations as they evolve as a better choice due to the attributes of cloud storage, flexible application development, data aggregation, scalability and platform of multiple services. The deployed cloud services eliminate the false identification of failures, which not only increase the availability of machines for intended operations but also reduce the maintenance cost.

#### 3.1 Proposed Cloud-based condition monitoring model: a layered approach

The proposed cloud services for machine vibration monitoring using IoT based framework have been modelled as a layered architecture as shown in **Figure 1** and implemented on the experimental set up as shown in **Figure 2**. The main layers [16] of the proposed model are the Platform as a Service (PaaS) layer, which is Google Cloud Platform, Software or Application as a Service (SaaS) layer where all the proposed services are deployed and Storage as a Service layer where all the data stores have been maintained. In online condition monitoring applications, the sensors are used to acquire vibration signals and communicated to the cloud storage for further processing. The proposed framework will enhance the machine condition monitoring functionality with methodologies of scalable and platform independent



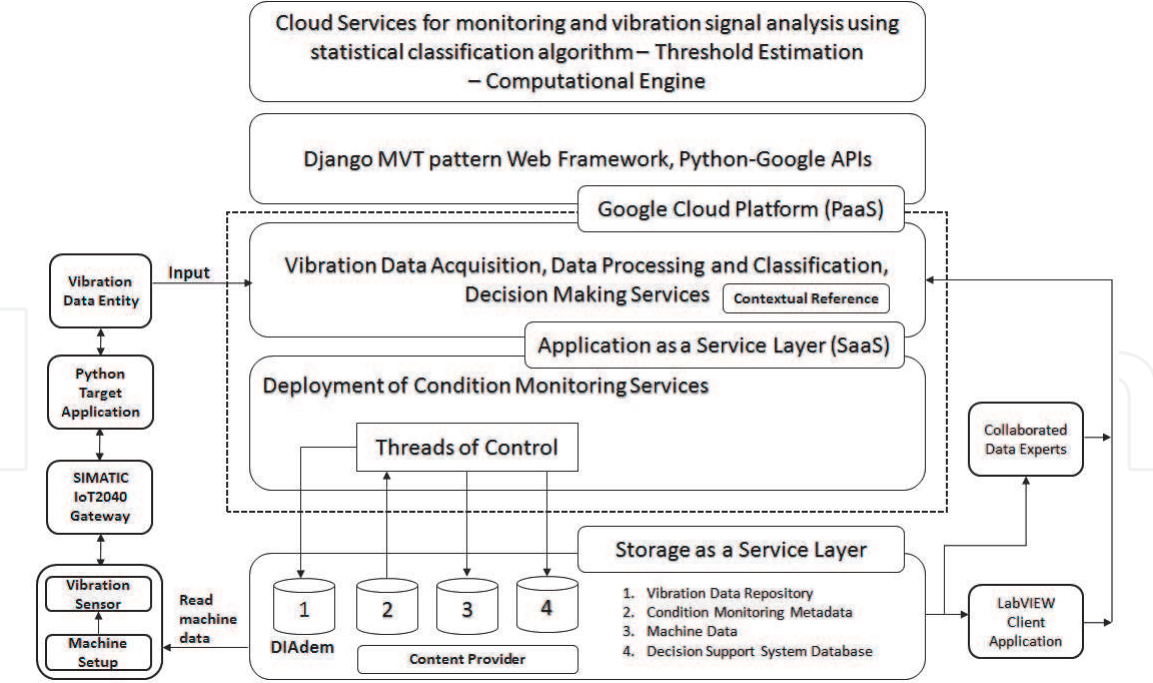


Figure 1.  
Cloud Services - Layered Architecture for Condition Monitoring Model.

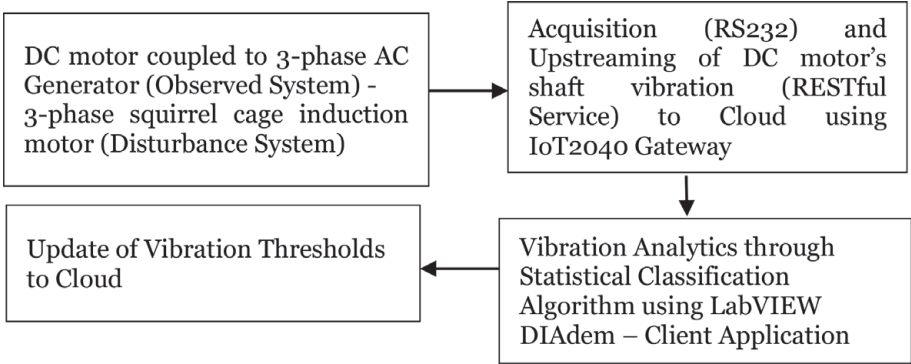


Figure 2.  
Experimental Setup.

data aggregation and collaborative analysis that the real-time industrial applications demand extensively.

The cloud infrastructure provides the fundamental resources needed to share upper level platforms and services. The “Storage as a Service” layer (Model perspective) and the physical resources needed for the “Computational Engine” and for communication among services along with core services for condition monitoring (View perspective) form the basis for delivering Infrastructure as a Service (IaaS). Google App Engine is used as Platform as a Service (PaaS) in the proposed model that provides a conducive environment to implement the cloud services. Google Cloud provides a set of APIs to aid the interaction between cloud components and end user applications, to enhance scalability, and ease deployment and management.

The Django Web Framework provides an environment to deploy the services developed in accordance with Model-View-Template pattern and the “google-cloud” API for Python provides interfaces for interacting with Google App Engine, which is configured as Python based cloud runtime environment. Software as a Service layer (SaaS) is for end users i.e., for the application developers. The services in this layer are typically accessed through Web portals using Templates. The



condition monitoring services provided with this model are normally referred to as SaaS and implemented in Views of Django Web Framework. The content providers of various services using a separate thread of control for each service, which have been developed by the registered end users, i.e., participating industries have stored the data in different formats under heterogeneous environment. The server application in the cloud environment has the control to manage the databases maintained in the Storage as Service layer. With the App Engine, each participating industry can write their application code, test it on their local machines and then deploy on cloud environment.

### 3.2 Creating, uploading and registering Cloud services for condition monitoring

Google App Engine is used for registering, uploading and accessing the condition monitoring services in the proposed cloud-based model for estimation of adaptive thresholds among various machines. The App Engine Software Development Kit (SDK) for Python is used to create and to link the services to the cloud. The location independent “google-cloud” client API as well as the shell SDK are providing the Python based cloud interfaces which can be accessed by the Django Web Services framework and Internet browser. The cloud applications need a configuration file i.e., “app.yaml” to deploy and run the application. The condition monitoring and decision making services configured using App Engine are easy to build, easy to maintain, and easy to scale as traffic and data storage needs grow. The services have been uploaded for ready to serve to any of number of machines located at distributed industrial environments.

### 3.3 Configuring condition monitoring services in Django Web framework

The Django “admin” is used to create a project named “ConditionMonitoring” using the command “django-admin startproject” with an application registered as “MonitoringApp”. The python script, “settings.py” defined within the project enumerates the default backend database and all the registered applications as detailed below:

```
#settings.py
DATABASES = {'default': {'ENGINE': 'django.db.backends.mysql',
                        'NAME': 'VIBRATION_DB', 'USER': 'username',
                        'PASSWORD': 'password', 'HOST': '192.168.1.112', 'PORT':
                        '5000'},}
INSTALLED_APPS = ['django.contrib.admin', ... .., 'MonitoringApp', ]
```

The python script, “manage.py” defined within the project starts the Webserver, migrates and synchronizes and flushes the databases if required. The script, “urls.py” defines the URL patterns to link the Views. The Views render the request/response to the Templates as XML for processing and HTML for presentation. The application, “MonitoringApp” describes and defines the required services in the Views. In the proposed cloud framework, the Model represents the Storage as a Service layer (virtual storage), the View represents the Application as a Service layer (SaaS) and the Template represents the presentation tier which includes Computational Engine.

### 3.3.1 Model

The logical schema for various database tables used in real-time condition monitoring to store the machine data, repository for previous decisions, maintenance schedules, data pertaining to historical conditions (i.e., decision support system) are implemented as Model entities under Django Web Framework, where each Model maps to a single database table dynamically. Model is the single definitive source of information about the data and it is defined in the “models.py” script. The metadata of the vibration signals acquired for condition monitoring of every machine, the metadata for the specification of machines of all participating industries and the metadata used for “Decision Support System” are described in the Model as follows:

**Vibration Signals Metadata**

[Component Name, Operating Condition, Disturbance Nature, Input Current, Vibration, Samples]

**Machines Metadata**

[Machine\_ID, Machine\_Type, Rated Voltage, Rated Current, Rated Speed]

**Decision Support System Database (Repository)**

[Machine\_ID, No. of Classes, Range of Classification, Class Width, Total Oscillations, Upper Threshold Class Cluster, Lower Threshold Class Cluster, Excess Positive Slopes, Excess Negative Slopes]

### 3.3.2 View

In general, the View retrieves data according to the path parameters defined in the “URL patterns” list, loads a template, renders the template with the retrieved data and returns the HTTP Response instance as output. Each View is a python service and an appropriate View is chosen by examining the URL that is requested as per the configurations made in the URL patterns. The codes pertaining to acquisition of vibration data, statistical classification algorithm, threshold estimation etc., are defined as python functions in the View’s sub-directory of the application.

The vibration data of the DC motor in the specially created experimental setup have been acquired when started at no load condition as well as loaded by AC Generator at fixed load changes. It is well known that the factory floor generally has lot of machines running together. Thus to create a similar field condition of the factory floor, an additional motor was installed nearer to the DC motor in the experimental setup and the shaft vibration data are again acquired for the stated conditions. The acquired vibration data under the operating conditions of starting to no load speed with and without external disturbance and loading are streamed to cloud through IoT2040 gateway. The LabVIEW client application enables collaborated real-time condition monitoring of any machine by integrating the non-stationary vibration analysis algorithm with a cloud service. The analysis results updated to the decision-making service lead to effective condition monitoring and make the maintenance of other connected devices/machines automatic and perfect scheduling. The updated results create contextual vibration references for assessing the condition of any other machine of same type that has been exposed to similar operating conditions.

### 3.3.3 Templates

A project can be configured with one or several template engines (<https://media.readthedocs.org>). Django defines a standard API for loading and rendering templates regardless of the backend. Loading consists of finding the template for a

given identifier and pre-processing it, rendering means interpolating the template with context data and returning the resulting string. The “templates” sub-directory should be created within the application directory by the end user manually. The template “dataframe1.html” to receive the response from the “ConditionMonitoring\_StandAlone” service is defined as follows:

---

```
<html>
<body bgcolor="#bg99FF">
<p>Type:{{Machine_Type}}</p>
<p>AcquisitionDevice:{{ SensorList }}</p>
<p>Machine_ID: {{Machine_ID}}</p>
<p>OperatingCondition:{{ Disturbance / Standalone / loading }}</p>
<p> DisturbanceNature: {{Constant Speed}}</p>
<p> Vibration: {{ samples .tdms }}</p>
<p>Speed:{{RatedSpeed}}</p>
</body>
</html>
```

---

The corresponding URL patterns entry is mentioned in the “urls.py” as follows:

---

```
from django.contrib import admin
from django.urls import path
from import .views
urlpatterns = [
    path('admin/', admin.site.urls),
    path('dataAquisition/',views.acquireVibrationData),
    path('dataframe1/',views.ConditionMonitoring_Standalone),
    path('dataframe2/',views.ConditionMonitoring_Disturbance),
    patch('dataframe3/',views.ConditionMonitoring_Loading),
    path('analysis/',views.StatisticalAnalysis),
    ... ..]
```

---

To communicate between various services, the data acquisition as well as estimated threshold values have been generated as XML for which the corresponding schemas are defined in the View’s subdirectory. The XMLized representation of the dataframe generated by the ConditionMonitoring\_Standalone service is given below:

---

```
<!--Dataframe representation-->
<?xml version="1.0"?>
<Dataframe>
  <Machine_Type>DC Motor</Machine_Type>
  <Machine_ID>EE-M5864</Machine_ID>
  <OperatingCondition>Standalone – Starting to No load speed
</OperatingCondition>
  <DisturbanceNature>NIL</DisturbanceNature>
  <Vibration>snl.tdms</Vibration>
  <Speed>1500</Speed>
</Dataframe>
```

---

In accordance with the response of the “ThresholdEstimation” service, the “DecisionMaking” service generates appropriate maintenance schedules in the XML form and helps in segregating false identification of failure status. If the attributes

of any data store have been changed, the corresponding Model updates the entries in the XML representation dynamically through its Views.

### **3.4 Deployment of vibration analysis based condition monitoring on Google Cloud platform**

Upon testing the Condition Monitoring Django Application in the local machine with the local SQL server running in the backend, the application is deployed in the Google App Engine Standard Environment (<https://cloud.google.com>) [17]. For deploying the application in the Cloud, a Google Cloud Platform (GCP) project is created in the GCP console. Further, the respective operating system's compatible Google Cloud SDK Shell is installed in addition to the Python Google-Cloud client libraries in the local development environment. The APIs are enabled in the GCP console and the Cloud SQL Proxy, which provides a secure access to the Cloud SQL is also installed with respective to the operating environment. The deployment of the application in Google Cloud is explained in the sequence of steps mentioned below:

#### *3.4.1 Creation and Initialization of Cloud SQL Instance*

Google Cloud provides Cloud SQL as Storage as a Service (SaaS), which supports all the database transactions with respect to the application to be deployed in the Cloud, which have been controlled by user defined threads of control.

#### *3.4.2 Configuring the application with the Google Cloud*

The configurations for the Database dictionary in "settings.py" with MySQL are set accordingly as per the Cloud SQL Instance connection name, database user, password and port.

#### *3.4.3 Execution of the application in the local development environment*

Before deploying the application, verification of the same is carried out in the local development environment by following the standard Web application execution procedure defined by Django Web Framework. The migrations are also carried out to set up the Models.

#### *3.4.4 Deploying the application on the Google App Engine Standard Environment*

All the static files of the application are gathered into a single directory by executing the command "python manage.py collectstatic". These static files are moved to the production site while deploying the application to the Google Cloud. The "requirements.txt" file is created to mention the dependencies and "app.yaml" which contains the environment, runtime and entry point is also created and finally, the application can be deployed on to the Google Cloud by executing the command "gcloud app deploy". Threads of control are introduced to populate the content provider by initiating the services in parallel mode to gather vibration data, machine data, to analyse the vibration signal data and to store the historical information in the "Decision Support System" data store for validation and further processing.



### 3.5 Statistical classification algorithm: computational engine

A new vibration analysis algorithm developed with statistical classification and clustering extracts the detailed signal features and estimates adaptive thresholds [14]. This signal decomposition algorithm, which forms the base for computational engine of the proposed IoT based cloud services model tracks the vibration signal transitions between the classes at multiple levels of amplitudes and computes the number of oscillations between the levels. While classifying the signal, the amplitude is segmented into a desired set of classes of equal and desired width. A transition matrix with 'n' number of classes and signal transitions is formulated out of the statistical classification of the data read from the cloud and streamed to DIAdem [18] as ".tdms" file by client application developed in LabVIEW platform for computation of the oscillation nature of the machine vibration.

In the proposed vibration analysis technique, the number of signal transitions has been considered to calculate the oscillations between every class and other classes to extract the threshold levels of vibration. The transition of the signal from a lower class to higher class and vice-versa are accounted as positive and negative slope respectively in the transition matrix. The analysis has been carried out by considering the transition matrix and progressing through the upper diagonal matrix row-wise and lower diagonal matrix column-wise elements or its vice-versa. The 'n' column vectors of upper diagonal matrix give the signal transitions from a class corresponding to the row to class of the diagonal element and 'n' row vectors of the lower diagonal matrix represent the signal transitions from class of the diagonal element to lower class corresponding to the column. The proposed algorithm determines the oscillations in the real time non-stationary vibration signal at multiple class levels using statistical classification of the signal amplitude as well as the transition matrix and delivers the features as Oscillation Matrix. The dominant classes of the Oscillation Matrix having higher count of oscillations with that of lower classes have been clustered together and identified as upper and lower threshold class clusters.

This algorithm is validated on the vibration signals acquired by IoT2040 gateway which is integrated with Google Cloud Platform and offered as cloud service in real time while testing cloud based condition monitoring. The computational engine has been integrated with the signal analysis to add more investigations towards the changes happening in the vibration during different operating conditions and offer adaptive thresholds for condition monitoring.

### 3.6 Integration and evaluation of the Cloud services based vibration analytics algorithm

To realize the effectiveness of the vibration thresholds estimated from the IoT based data analysis, a comparative analysis has been made with the results of vibration data acquired through myRIO. The same experimentation has been carried out for acquiring the shaft vibration data using myRIO1900 [19] as acquisition device and tri-axial accelerometer (ADXL345) as vibration sensor. The data acquired in both cases have been fed to the computational engine for carrying out statistical classification analysis for fixing the clusters of vibration thresholds precisely at the following dynamic operating conditions:

- Starting to no load speed and load changes in standalone mode of operation
- Starting to no load speed with induced external disturbance

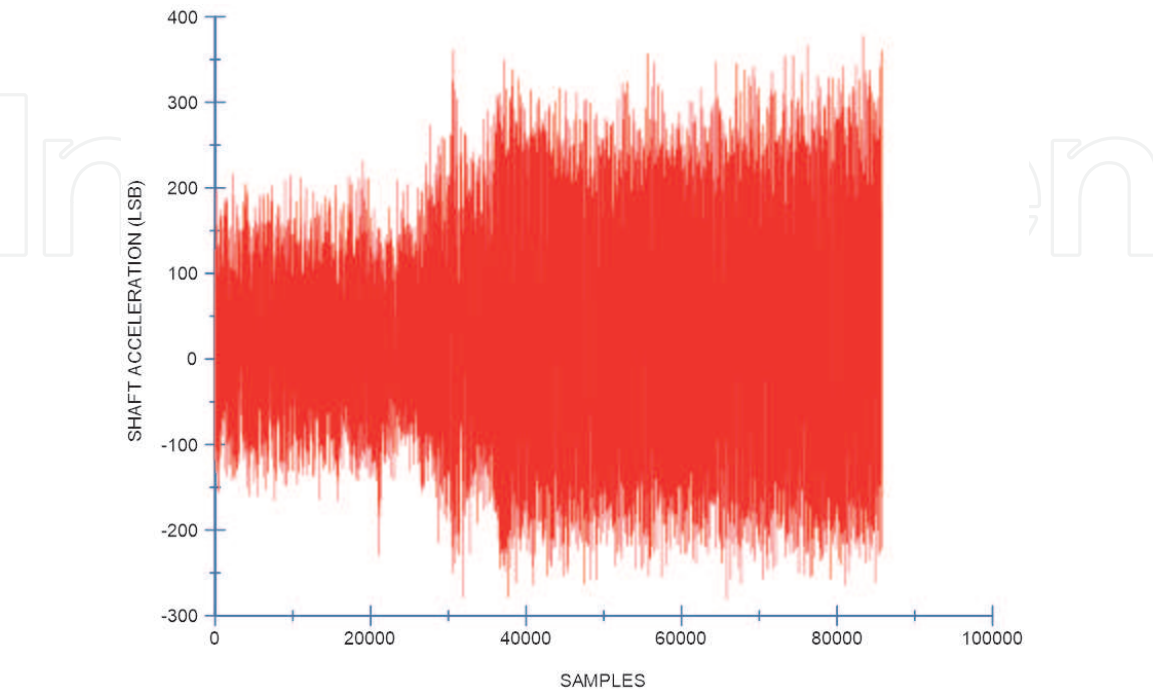
In all test cases and operating conditions, the single valued vibration threshold is replaced with a cluster of upper and lower thresholds for safe operation of a machine. This has been implemented on 3,96,000 samples of vibration signal acquired from the DC motor under the stated conditions. The acquired data are logged in an excel file and imported to the computational engine in which the statistical classification technique is implemented using NI DIAdem [18]. This technique has been tested on the DC motor shaft vibration signals at the above stated operating conditions using different classification criteria. In each case, the same vibration signal is segmented to study about the oscillations pattern.

The shaft vibration signal of DC motor pertaining to the operating conditions of starting to no load speed (standalone and disturbance conditions) and loading (standalone condition) have been acquired by myRIO application developed using LabVIEW FPGA and RT programming through the tri-axial accelerometer ADXL345 having sensitivity of 256 LSB/g. The vibration data during the same operating conditions are acquired by IoT2040 gateway [20] from a Piezo electric sensor through serial interface under the measurement unit of ‘g’ and transferred the scaled value in LSB to cloud simultaneously. The upper and lower threshold class clusters are identified with the application of signal processing algorithm on the vibration signal and the non-stationary vibration data corresponding to different operating conditions are characterized using the transition matrix obtained from the statistical classification.

3.6.1 From Starting to No Load Speed – Standalone Condition

The acquired data pertaining to this operating condition with respect to myRIO and IoT Gateway (as shown in **Figure 3**) applications have been fed to the computational engine for statistical classification and analyzed with different classification configurations. The observed signal peak values and the configuration settings assumed for classification in each case are listed in **Table 1**.

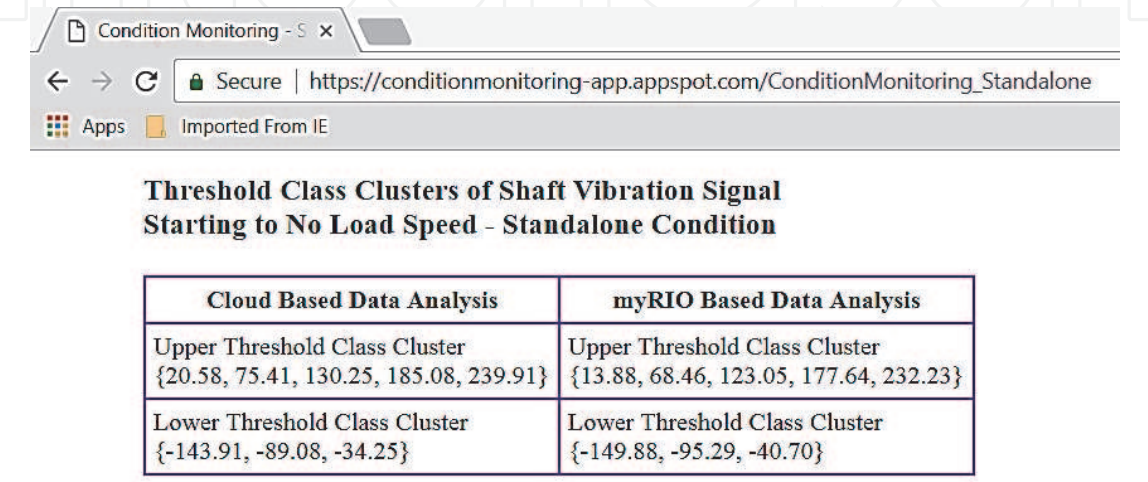
The vibration analysis is carried out by the computational engine and the estimated upper and lower threshold clusters are rendered to the “VibrationAnalysis”



**Figure 3.**  
*Shaft Acceleration Acquired by IoT Gateway – Standalone Condition.*

Observed Peak Values			Classification Configuration	
Maximum Amplitude	Minimum Amplitude	Range	No. of Classes	Class Width
myRIO				
368 LSB	-286 LSB	654	12	54.5
IoT Gateway				
377 LSB	-281 LSB	658	12	54.83

**Table 1.**  
*Measured Values and Classification Configuration - Standalone Starting to No Load Speed*



**Figure 4.**  
*Adaptive Threshold Class Clusters – Standalone Condition*

template, which can be accessed through the cloud environment by specifying the URL: “*https://conditionmonitoringapp-appspot.com/ConditionMonitoring\_Standalone*” and the results are displayed in the form of HTML as shown in the **Figure 4**. Thus, the occurrence of faults or any abnormality at this operating condition can be diagnosed precisely with the shift in the oscillation percentages of the denser class regions and threshold class clusters from the predetermined values.

3.6.2 *From starting to no load speed with external disturbance*

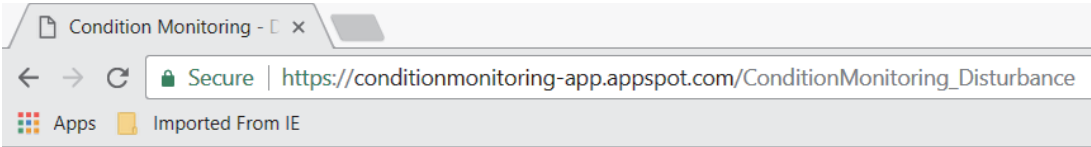
Similar to the standalone mode, the experiment has been carried out with an external disturbance using IoT and myRIO based data acquisition systems and the

Observed Peak Values			Classification Configuration	
Maximum Amplitude	Minimum Amplitude	Difference	No. of Classes	Class Width
myRIO				
453.6 LSB	-296.3 LSB	749.90	14	54.01
IoT Gateway				
460.19 LSB	-296.08 LSB	756.27	14	54.01

**Table 2.**  
*Measured Values and Classification Configuration - Starting to No Load Speed under External Disturbance.*

shaft vibration pattern has been examined. While performing analysis on the vibration signals with mechanical disturbance, the observed peak values of the vibration signal and the configuration settings made for analysis are listed in **Table 2**.

Using the resulted transition matrix, the oscillations existing between every class and its lower classes are calculated and the dominant classes with more percentage of oscillations measured during the presence of external disturbance have been identified to form the upper threshold class cluster. To form the lower threshold class cluster, every class of the upper threshold class cluster that has made 65 percent or more number of oscillations cumulatively with its lower classes are considered and the results are rendered to the “Disturbance Condition” template, which can be accessed through the cloud environment by specifying the URL: *https://conditionmonitoringapp-appspot.com/ConditionMonitoring\_Disturbance* and the results are displayed in the form of HTML as shown in **Figure 5**. There are changes in the vibration pattern due to the external disturbance and it is observed from the results shown, the fact of fixing adaptive threshold for a machine when exposed to external disturbances at a particular operating condition. The severity of the disturbance can be observed by measuring the range of shifts from the limits of the upper and lower threshold class clusters estimated during standalone condition.



**Threshold Class Clusters of Shaft Vibration Signal  
Starting to No Load Speed - Disturbance Condition**

Cloud Based Data Analysis	myRIO Based Data Analysis
Upper Threshold Class Cluster {1.03, 55.05, 109.06, 163.08, 217.10}	Upper Threshold Class Cluster {-1.68, 51.88, 105.46, 159.03, 212.61}
Lower Threshold Class Cluster {-161.03, -107.01, -52.99}	Lower Threshold Class Cluster {-162.41, -108.84, -55.26}

**Figure 5.**  
*Adaptive Threshold Class Clusters – Disturbance Condition.*

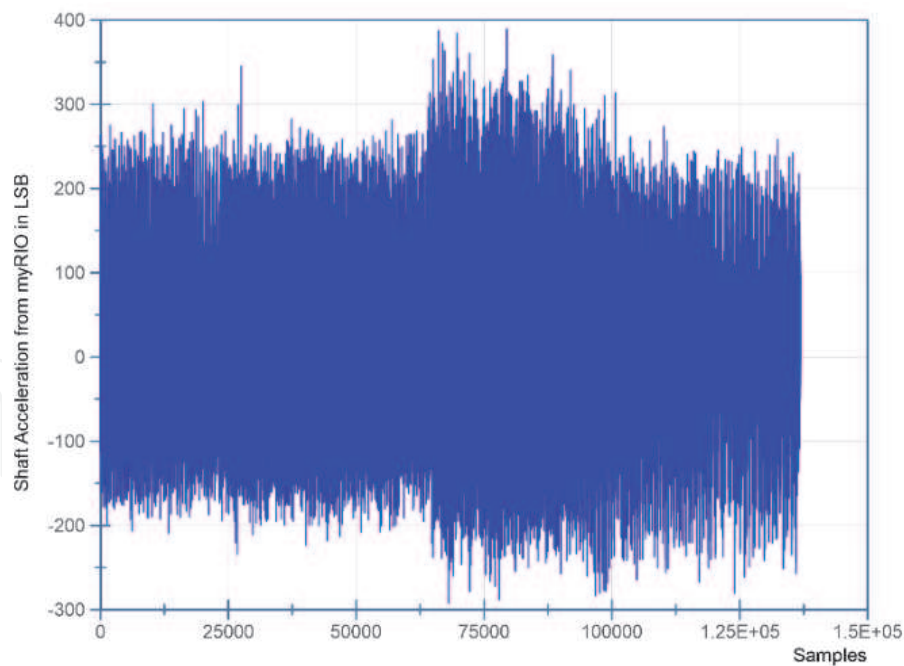
3.6.3 Load Changes at Standalone Condition

The shaft acceleration i.e., the vibration signals acquired by the myRIO and IoT based cloud applications during the load changes made at standalone running condition of the DC machine are shown in **Figures 6** and **7**. The observed characteristics and analysis settings used in statistical classical algorithm for condition monitoring analysis in both cases are presented in the following **Table 3**.

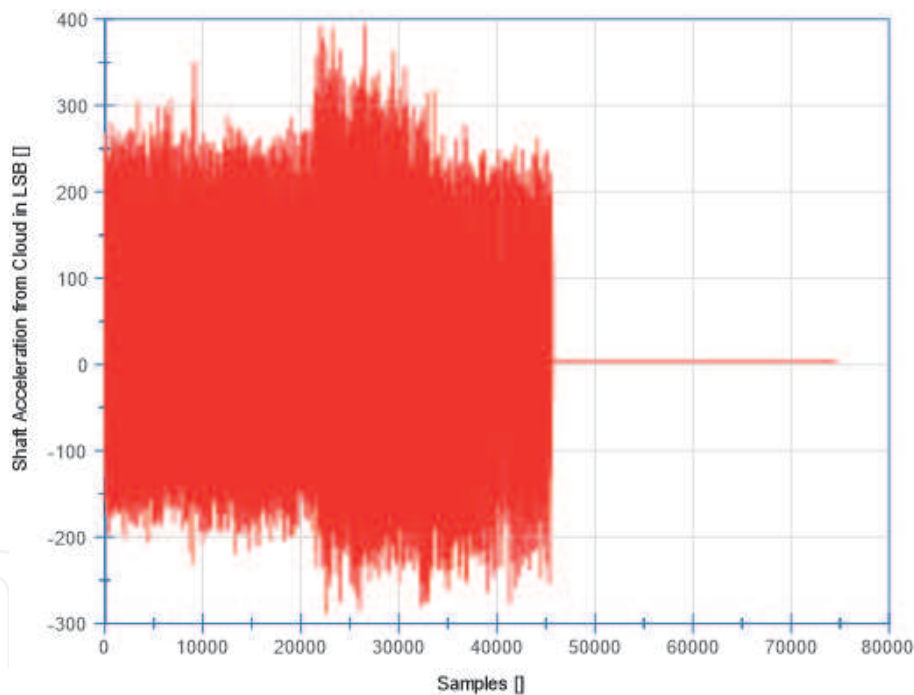
The results of upper and lower threshold class clusters obtained from the implementation of the signal decomposition algorithm are rendered to the “LoadingCondition” template, which can be accessed through the cloud environment by specifying the URL: “*https://conditionmonitoring-app.appspot.com/ConditionMonitoring\_Loading*” and the results are displayed as shown in **Figure 8**.

The analysis carried out using the statistical classification based signal decomposition technique for different machine operating conditions is based on the machine vibrations occurring within the permissible limits. The consistency in the





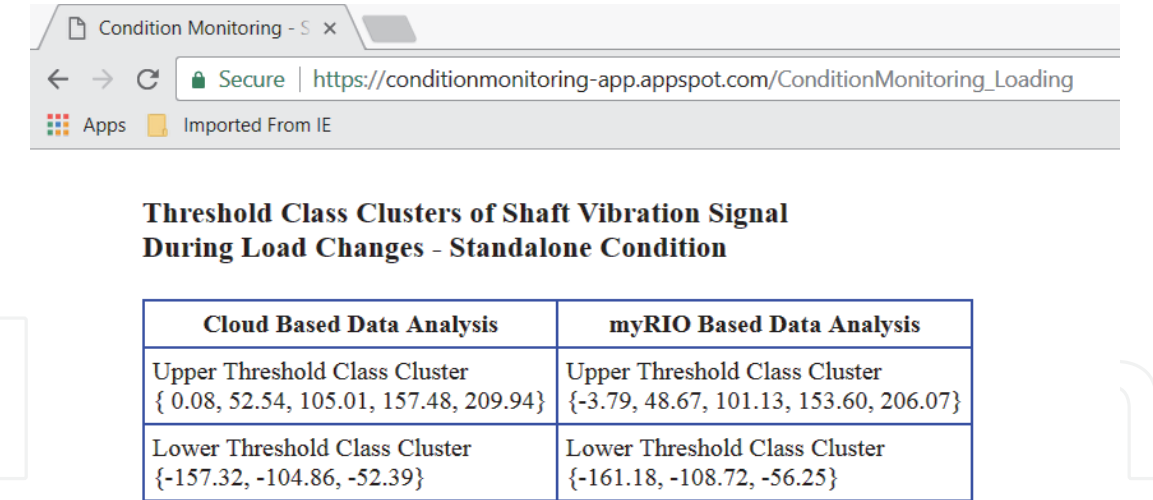
**Figure 6.**  
*Shaft Acceleration acquired by myRIO during Loaded Condition.*



**Figure 7.**  
*IoT based acquisition of Shaft Acceleration during Loaded Condition.*

Observed Peak Values			Classification Configuration	
Maximum Amplitude	Minimum Amplitude	Difference	No. of Classes	Class Width
myRIO				
389 LSB	-292 LSB	681.00	13	52.3
IoT based Cloud Application				
393.51 LSB	-288.49 LSB	682.00	13	52.4

**Table 3.**  
*Measured Values and Classification Configuration – Load Changes.*



**Figure 8.**  
*Adaptive Threshold Class Clusters – Loaded Condition.*

total number of oscillations and marginal deviations in the threshold class clusters reveal the flexibility of choosing any of the desired classification criteria during real time implementation. Contrary to the currently adhered thresholds, the adaptive cluster based thresholds possess the significance of tracking the condition of the machine without demarcating the naturally occurring vibration variations as crossing limits during condition monitoring. Hence, this technique avoids false identification of failures caused due to incorrect thresholds and also identifies the ignored failures. This cloud based analysis of random vibration signal is a perception that has been believed to offer better alternative for deriving decisions for efficient condition monitoring when other analysis techniques find challenges in bringing out precise and faster solutions for condition monitoring under dynamic conditions.

3.7 Results and discussion

The DC motor shaft vibration pattern has been examined by acquiring the vibration signal through IoT2040 gateway using Python interface and myRIO using LabVIEW interface considering the same machine operating conditions. The vibration signals acquired by various data acquisition devices have been analyzed by the statistical classification based signal decomposition algorithm considering three different modes of application platform as given below:

- i. An independent vibration analysis application in LabVIEW platform (myRIO based analysis)
- ii. LabVIEW application integrated with IoT service (IoT based analysis)
- iii. LabVIEW application hosted as a cloud service in Google Cloud Platform (Cloud based analysis)

The upper and lower threshold class clusters of DC motor’s shaft vibration determined using the above applications are furnished in **Table 4** which define the scope of the amplitude levels between which majority of the shaft vibrations oscillate during the specified operating conditions. In either case of analysis, the proposed algorithm has uniformly brought out the changes that had happened in the vibration pattern and upholds the fact of fixing thresholds adaptive to the operating

Starting to No Load Speed (Standalone Condition)			Starting to No Load Speed (Disturbance Condition)			Loading (Standalone Condition)		
Cloud Based Analysis	IoT Based Analysis	myRIO Based Analysis	Cloud Based Analysis	IoT Based Analysis	myRIO Based Analysis	Cloud Based Analysis	IoT Based Analysis	myRIO Based Analysis
Upper Threshold Class Cluster								
{	{	{	{	{	{	{	{	{
20.58	15.20	13.88	1.03	0.03	-1.68	0.08	-0.76	-3.79
75.41	69.79	68.46	55.05	53.67	51.88	52.54	52	48.67
130.25	124.37	123.05	109.06	107.32	105.46	105.01	104.76	101.13
185.08	178.95	177.64	163.08	160.96	159.03	157.48	157.53	153.60
239.91	233.54	232.23	217.10	214.60	212.61	209.94	210.30	206.07
}	}	}	}	}	}	}	}	}
Lower Threshold Class Cluster								
{	{	{	{	{	{	{	{	{
-143.91	-148.54	-149.88	-161.03	-160.89	-162.41	-157.32	-159.07	-161.18
-89.08	-93.95	-95.29	-107.01	-107.25	-108.84	-104.86	-106.30	-108.72
-34.25	-39.37	-40.70	-52.99	-53.60	-55.26	-52.39	-53.53	-56.25
}	}	}	}	}	}	}	}	}

**Table 4.**  
*Adaptive Threshold Clusters Identified by Different Analysis Methods.*

Starting to No Load Speed (Standalone Condition)				Starting to No Load Speed (Disturbance Condition)				Loading (Standalone Condition)			
Cloud Based Analysis		IoT Based Analysis		Cloud Based Analysis		IoT Based Analysis		Cloud Based Analysis		IoT Based Analysis	
D	%	D	%	D	%	D	%	D	%	D	%
	D		D		D		D		D		D
6.7	1.0	1.32	0.2	2.71	0.4	1.71	0.2	3.87	3.03	0.6	0.4
7.0	1.1	1.33	0.2	3.17	0.4	1.79	0.2	3.87	3.33	0.6	0.5
7.2	1.1	1.32	0.2	3.6	0.5	1.86	0.2	3.88	3.63	0.6	0.5
7.4	1.1	1.31	0.2	4.05	0.5	1.93	0.3	3.88	3.93	0.6	0.6
7.7	1.2	1.31	0.2	4.49	0.6	1.99	0.3	3.87	4.23	0.6	0.6

**Table 5.**  
*Deviations of Adaptive Threshold Clusters of Same Operating Condition from myRIO Based Analysis.*

condition. The values of deviation between the deduced threshold levels considering the same and various operating conditions are enlisted in **Tables 5** and **6**. The margin of deviations between the threshold class clusters specific to an operating condition determined using IoT [14] and cloud based analysis methods with reference to the amplitude range of vibration signal measured using myRIO (**Table 5**) are observed to be negligible as compared to the deviation between threshold class clusters identified by any particular analysis method for different operating conditions (**Table 6**). In each of the cases considered, the incipient faults or abnormalities during any of the operating conditions can be diagnosed precisely by analysing the margin of deviations in the threshold class clusters. The difference in the deviation values observed from **Tables 5** and **6** reveals that the threshold class clusters obtained by the analysis of data acquired either from IoT device, cloud

Disturbance Condition (Deviation D)			Loading at Standalone Condition (Deviation D)		
Cloud Based Analysis	IoT Based Analysis	myRIO Based Analysis	Cloud Based Analysis	IoT Based Analysis	myRIO Based Analysis
20	15	16	21	16	18
20	16	17	23	18	20
21	17	18	5	20	22
22	18	19	28	21	24
23	19	20	30	23	26

**Table 6.**  
*Deviations of Threshold Class Clusters Between Different Operating Conditions for specific method of analysis*

service or from myRIO device do not lead to incorrect decisions and tends to recognize the change of operation conditions without ambiguity.

In spite of some minor deviations in the threshold values determined for every operating condition during different methods of analysis, the substantial differences in the threshold class clusters identified due to the change of operating conditions in all the analysis methods validate also the fact that the incipient faults or abnormalities (during any of the operating condition or change in the operating conditions) can be diagnosed without ambiguity using the proposed statistical classification algorithm integrated with the cloud environment.

In this work, the actual machine condition is monitored in online mode by continuous acquisition of vibration signals and simultaneous estimation of the threshold levels at different operating conditions using developed cloud services of data acquisition and data processing. The machine’s vibration signature will not change over a period of time until some disturbance or loading has occurred. During abnormal conditions, the vibration amplitude may increase or decrease from the normal value. Integrating the actual deviation from the mean values of the upper and lower threshold clusters with historical data for the same or similar machine under the same operating condition facilitates to evaluate the current condition of the machine and hence to schedule suitable maintenance action.

It is observed from the analysis that the vibration threshold class clusters identified for any particular operating condition cannot be maintained as an alarm for monitoring of machines at all conditions to decide appropriate maintenance schedules. Further the analysis reveals the fact, a neighboring machine running at constant speed of 1500 rpm creates a vibration effect equivalent to that of loading in other machines which substantiates the fixation of the operational limits for mechanical and electrical loading, speed and torque capacity etc., during machine maintenance.

The consolidated results obtained from the implementation of the proposed algorithm under unequal and equal classification criteria in local and cloud platforms illustrate the reliability of the analysis technique in yielding consistent results. As illustrated above, the characteristic of the technique to extract the signal features in such a way that they identify the unremarkable disturbances in the operating conditions, and effects of change of operating conditions enable fixation of cluster based thresholds adaptive to machine operating conditions rather than fixed and ambiguous threshold levels. Moreover, the consistency of the observed threshold values under any of the classification parameters ensures the efficiency of the technique. The technique employed on shaft vibration signal at different operating



modes has identified ignored disturbances, intensity and characteristics of such disturbances and incipient changes in the operational behavior both in the on premise analysis and cloud based analysis. Thus the proposed IoT based cloud deployment will help to prescribe the operational constraints for machines in real time applications so that the machine can deliver improved performance and have extended lifetime.

4. Conclusion

In either case of analysis based on myRIO, IoT device or Cloud based, the investigation uniformly brings out the changes that had happened in the vibration pattern and upholds the fact of fixing thresholds adaptive to the operating condition. The integration of vibration sensors and actuators through Python and LabVIEW interfaces with cloud in real time ascertains generic, interoperable and ubiquitous computational nature of the model for implementation of effective condition monitoring. In this research, the convergence of cloud and IoT technologies for analysis of real-time systems has been brought into implementation for condition monitoring of electrical machines to support scalable and interoperable data exchange with features of flexible and collaborative analytics, fixation of adaptive alarms with contextual thresholds and control of multiple machines in real-time operating environment. The model gives collaborative access to machine data from any geographical location for analysis and decision making.

In summary, the cloud-based vibration monitoring model implemented in Google Cloud Platform offers services for

- Data acquisition from the sensors mounted on the shafts of the DC motors.
- Data storage to store the enormous amount of acquired signal data from multiple sensors.
- Data classification of vibration signals for effective statistical analysis to estimate adaptive cluster of thresholds and
- Decision making for condition assessment.

These services have been offered on demand over the Internet to utilize the reliable service of the machines in a persistent way. The computational engine, which is included in the model performs inherent statistical analysis of the vibration signals to estimate the cluster of thresholds adaptive to various operating conditions. The services have been deployed without any limitation in a cloud environment and the industrial applications can share information using the deployed services from anywhere on demand basis. The threshold values estimated using cloud services are compared with that of the vibration analysis carried out on the machine beds locally using myRIO for data acquisition in LabVIEW ensures the integrity of the cloud-based model with assured scalability.

List of nomenclature

IoT	Internet of Things
RIO	Reconfigurable Input Output
ISO	International Organization for Standardization

SCADA	Supervisory Control and Data Acquisition
PLC	Programmable Logic Controller
VPN	Virtual Private Network
LabVIEW	Laboratory Virtual Instrument Engineering Workbench
GCP	Google Cloud Platform
MVT	Model-View-Template
PaaS	Platform as a Service
SaaS	Software as a Service
IaaS	Infrastructure as a Service
API	Application Programming Interface
.tdms	Technical Data Management Streaming format
SDK	Software Development Kit
HTTP	Hypertext Transfer Protocol
HTML	Hypertext Markup Language
URL	Uniform Resource Locator
XML	eXtensible Markup Language
SQL	Structured Query Language
SDK	Software Development Kit
FPGA	Field-Programmable Gate Array
RT	Real Time
g	Acceleration due to gravity (9.81 m/s <sup>2</sup> )
LSB/g	Least Significant Bit per g
% D	Percentage Deviation

### Author details

Ganga Dhandapani<sup>1\*</sup> and V. Ramachandran<sup>2</sup>

1 Department of Electrical and Electronics Engineering, NIT Nagaland,  
Dimapur, India

2 Department of Computer Science and Engineering, DMI College of Engineering,  
Chennai, India

\*Address all correspondence to: [gangaadhan@gmail.com](mailto:gangaadhan@gmail.com)

### IntechOpen

© 2021 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

## References

- [1] Diego Galar, Peter Sandborn, Uday Kumar, Carl-Anders Johansson, S.: SMART: Integrating Human Safety Risk Assessment with Asset Integrity. In: Giorgio Dalpiaz, Gianluca D'Elia, Riccardo Rubini, Marco Cocconcelli, Fakher Chaari, Mohamed Haddar, Radoslaw Zimroz, Walter Bartelmus (eds.), *Proceedings of the Third International Conference on Condition Monitoring of Machinery in Non-Stationary Operations CMMNO 2013*, LNME, Springer; 2014. p. 37–59
- [2] Kirubashankar, R., Krishnamurthy, K., Indra, J., Vignesh, B: Design and Implementation of Web Based Remote Supervisory Control and Information System. *International Journal of Soft Computing and Engineering (IJSCE)* ISSN: 2011:1-4:2231-2307
- [3] Larry Combs: Cloud Computing for SCADA. White Paper from InduSoft.2011
- [4] Omid Givehchi, Henning Trsek, Juergen Jasperneite: Cloud Computing for Industrial Automation Systems - A Comprehensive Overview. In: *Proceedings of the IEEE 18<sup>th</sup> Conference on Emerging Technologies and Factory Automation*; 2013.p.1-4.
- [5] Omid Givehchi, Juergen Jasperneite: Industrial Automation Services as Part of the Cloud: First Experiences. In: *Proceedings of the Jahreskolloquium Kommunikation in der Automation – KomMA*; 2013.
- [6] Omid Givehchi, Jahanzaib Imtiaz, Henning Trsek, Juergen Jasperneite: Control-as-a-Service from the Cloud: A Case Study for using Virtualized PLCs. In: *Proceedings of 10<sup>th</sup> IEEE Workshop on Factory Communication Systems*; 2014.p.1-4
- [7] Francisco Macia-Perez, Jose Vicente Berna-Martinez, Diego Marcos-Jorquera, Iren Lorenzo-Fonseca, Antonio Ferrandiz-Colmeiro: A New Paradigm: Cloud Agile Manufacturing. *International Journal of Advanced Science and Technology*. 2012:45:47-53.
- [8] Bureau of Indian Standards IS 12075, “Mechanical Vibration of Rotating Electrical Machines with Shaft Heights 56 mm and Higher - Measurement, Evaluation and Limits of Vibration Severity”, [ETD 15: Rotating Machinery], 2008
- [9] National Instruments, White Paper: Addressing Challenges of Fleetwide Monitoring. 2016. Available: [download.ni.com/evaluation/crio/24346\\_Fleetwide\\_Monitoring\\_WP\\_IA.pdf](http://download.ni.com/evaluation/crio/24346_Fleetwide_Monitoring_WP_IA.pdf) [Accessed: 2020-12-29]
- [10] Jim Montague. Flexibility Driving the Future of Automation – Flexible Production, Cloud Computing Pave the Way to Future for Rockwell Automation and User, Automation Fair - A special report. Control and control design [Internet]. 2012. Available : <https://www.controlglobal.com/articles/2012/pauto-fair-17/> [Accessed: 2020-12-29]
- [11] Advantech Technical White Paper. Cloud-Based SCADA as an IIoT Gateway. 2015. Available: <https://silotips/download/cloud-based-scada-as-an-iiot-gateway>, p.1-11 [Accessed: 2020-12-29]
- [12] Steve Lacey. From local to cloud-based condition monitoring. Quarry Management [Internet]. 2017. Available: <https://www.engineerlive.com/content/local-cloud-based-condition-monitoring> [Accessed: 2020-12-29]
- [13] Mallikarjun Kande, Alf J. Isaksson, Rajeev Thottappillil, Nathaniel Taylor: Rotating Electrical Machine Condition Monitoring Automation—A Review. *Machines*. 2017:5-4:24. DOI : <http://dx.doi.org/10.3390/machines5040024>

[14] D. Ganga, V. Ramachandran: IoT based Vibration Analytics of Electrical Machines. IEEE Internet of Things Journal. 2018:5-6:4538 –4549. ISSN: 2327-4662, DOI.: 10.1109/JIOT.2018.2835724

[15] D.Ganga, V. Ramachandran: Dynamic analysis of Vibration Signals and Adaptive Measures for effective Condition Monitoring of Electrical Machines. International Journal of Condition Monitoring and Diagnostic Engineering Management. 2018:21-2: 29-38. ISSN 1363 – 7681, 2018

[16] Balasingh Moses. M, Ramachandran. V, Lakshmi. P: Cloud Services for Power System Transient Stability Analysis. European Journal of Scientific Research. 2011:56-3:301-310.

[17] Google Cloud. Running Django on App Engine Standard Environment [Internet]. 2018. Available: <https://cloud.google.com/python/django/appengine> [Accessed: 2020-12-29]

[18] DIAdem 2017, Part Number 370858N-01, National Instruments [Internet]. 2017. Available: <http://zone.ni.com/reference/en-XX/help/370858N-01/> [Accessed: 2020-12-29]

[19] National Instruments. NI myRIO-1900 User Guide and Specifications [Internet]. 2013-2016. Available: <https://www.ni.com/pdf/manuals/376047c.pdf> [Accessed: 2020-12-29]

[20] Siemens. SIMATIC IOT SIMATIC IOT2020 SIMATIC IOT2040, Operating Instructions Manual - A5E37656492-AB [Internet]. 2016. Available: [https://support.industry.siemens.com/cs/attachments/109741658/iot2000\\_operating\\_instructions\\_e\\_en-US.pdf](https://support.industry.siemens.com/cs/attachments/109741658/iot2000_operating_instructions_e_en-US.pdf) [Accessed: 2020-12-29]