"Data Driven Framework for Real-Time Machine Learning Based Prediction on Engine Vibration Sensory Data."

A

Project Report

submitted in partial fulfillment of the requirementsfor the award of the degree of

MASTER OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND ARTIFICIAL NEURAL NETWROK

by

Name	Roll No.
Ojasvi Goswami	R102214004

Under the guidance of

Prof. Vishal Kaushik

Department of Computer Science & Engineering
Centre for Information Technology
University of Petroleum & Energy Studies
Bidholi, Via Prem Nagar, Dehradun, UK

April – 2016

The innovation driven
E-School

CANDIDATE'S DECLARATION

I hereby certify that the project work entitled "***Data Driven Framework for Real-Time Machine Learning Based prediction on Engine Vibration Sensory data***" in partial fulfillment of requirement for the award the degree of **MASTER OF TECHNOLOGY IN ARTIFICIAL INTELLIGENCE & ARTIFICIAL NEURAL NETWORK** and submitted to Department of Computer Science & Engineering at center for Information Technology, University of Petroleum & Energy Studies, Dehradun, in authentic record of my work carries out during a period from December, 2015 to April, 2016 under guidance the supervision of Prof. Vishal Kaushik

The matter presented in this project has not been submitted by me/ us for the award of any other degree of this or any other University.

Ojasvi

Ojasvi Goswami
R102214004

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: 08 / 04 / 2016

Prof. Vishal Kaushik
Project Guide

Dr. Amit Agrawal
Program Head –M.Tech Artificial intelligence & Artificial Neural Network.
Center for Information Technology
University of Petroleum & Energy Studies
Dehradun – 248 001 (Uttarakhand)

ACKNOWLEDGEMENT

I wish to express our deep gratitude to our guide **Prof. Vishal Kaushik**, for all advice, encouragement and constant support he has given us throughout our project work. This work would not have been possible without his support and valuable suggestions.

I am heartily thankful to my thesis advisor **MR. NRK Rao** and **Mr. Samit Prabhu** at **L&T Infotech**. The door to **Mr. NRK Rao** office was always open whenever I ran into a trouble spot or had a question about my research or writing.

I am heartily thankful to my course coordinator, Prof. Vishal Kaushik, for the precise evaluation of the milestone activities during the project timeline and the qualitative and timely feedback towards the improvement of the project.

I sincerely thank to our respected Program Head of the Department, Dr. Maneesh Prateek, for his great support in doing our project in **Area** at **CIT**.

I am also grateful **Dr. Kamal Bansal** Dean CoES, UPES for giving us the necessary facilities to carry out our project work successfully.

We would like to thank all our **friends** for their help and constructive criticism during our project work. Finally, we have no words to express our sincere gratitude to our **parents** who have shown us this world and for every support they have given us.

Ojasvi Goswami

R102241004

ABSTRACT

Vibration Analysis (VA), applied in an industrial or maintenance environment aims to reduce maintenance costs and equipment downtime by detecting equipment faults.VA is a key component of a Condition Monitoring (CM) program, and is often referred to as Predictive Maintenance (PdM). CBM dictates that maintenance should only be performed when certain indicators show signs of decreasing performance or upcoming failure.

Human specialists identify/detects unwanted behavior of an engine based on knocking and engine vibration. The project aims on engine's vibration data which is used to determine the condition of an engine and implement a predictive maintenance application. In the project we studied various clustering algorithm to identify the anomalies and the outliers, while classification algorithm for justifying the problem. The analog vibration consists of an Amplitude and Frequency; these provide the desired information for engine diagnosis. Data observations from sensors are streamed real time to perform analytics. Machine learning technique is used to analyze engine vibration for real time monitoring and predictive maintenance.

Table of Contents

LIST OF TABLES

List of Figure

1. INTRODUCTION

Condition based support (CBM) or Predictive Maintenance (PdM) can be best depicted as support rehearsed when need emerges. This is finished by observing the state of the machine constantly or intermittently relying on the requirement for the accessibility of the machine. The support is started when markers hint at the flaws in the beginning stages. In straightforward words, the principle rule is to keep up the right gear at the ideal time.

Predictive maintenance uses non-destructive monitoring during the normal operation of the equipment. Sensors installed on the equipment collect valuable data that can be used to predict and prevent failures. Current techniques for predictive maintenance include vibration analysis, acoustical analysis, infrared monitoring, oil analysis, and model-based condition modeling. Vibration analysis uses sensors such as accelerometers installed on a motor to determine when it is operating abnormally. Vibration analysis is the most widely used approach to condition monitoring, accounting for up 85% of all systems sold. Acoustical analysis uses sonic or ultrasound analysis to detect abnormal friction and stress in rotating machines. While sonic techniques can detect problems in mechanical machines, ultrasound is more flexible and can detect issues in both mechanical and electrical machines. Infrared analysis has the widest range of applications, spanning low to high-speed equipment as well as mechanical and electrical devices.

Vibration analysis is frequently used in technique to monitor machine condition for rotating machinery, because of comparatively faster than any other available techniques for data collection and its interpretation. Since the information is gathered as digitally examined time space signals, the vibration analysis method enable further changes by using computers.

Vehicle reliability and thus availability, or uptime, is increasingly important to haulers as FMS systems become more widespread. Reliability is the next area of improvement and the demand for less unplanned stops is driven by the fierce competition in haulage as most of the other parts of their business already is optimized. Reliability can be controlled by improving vehicle quality and by partially preventing the maintenance actions.

The sensory data generated are real time process. The observations are collected from six differently placed sensors i.e. axial, horizontal and vertical, with opposed functionalities i.e filtered and non-filtered. Machine Learning algorithm is applied to perform future prediction.

1.1 Benefits of vibration analysis and Predictive maintenance

- Minimizes or eliminates costly downtime - increases profitable uptime.
- Minimizes or eliminates catastrophic machinery failures - damage from catastrophic failure is usually much more extensive than otherwise would have been.
- Reduces maintenance costs.
- Reduces unscheduled maintenance - repairs can be made at times that least affect production.
- Reduces spare parts inventories - many parts can be purchased just in time for repairs to be made during scheduled machinery shutdowns.
- Optimizes machinery performance - machinery always operates within specifications.
- Reduces excessive electric power consumption caused by inefficient machinery performance - saves money on energy requirements.
- Reduces need for standby equipment or additional floor space to cover excessive downtime - less capital investment required for equipment or plant.

- Increases plant capacity.
- Reduces depreciation of capital investment caused by poor machinery maintenance - well maintained machinery lasts longer and performs better.
- Reduces unnecessary machinery repairs - machines are repaired only when their performance is less than optimal.
- Minimizes or eliminates the possibility that machinery repairs were the wrong repairs.
- Reduces the number of dissatisfied customers or lost customers due to poor quality - with less than optimal machine performance, quality always suffers.
- Reduces rework of goods caused by machines operating at less than optimal condition.
- Reduces scrap caused by poorly performing machinery.
- Reduces overtime required to make up for lost production due to broken down or poorly performing machinery.
- Reduces penalties that result from late deliveries caused be broken down or poorly performing machinery.
- Reduces warranty claims due to poor product quality caused by poorly performing machinery.
- Reduces the possibility of accepting recently purchased new or used machinery with defects - the invoice is not paid until the defects are corrected.
- Increases the likelihood that newly purchased new or used machinery meets specifications.
- Increases machinery safety - injuries are often caused by poorly performing machinery.
- Reduces safety penalties levied against a company for unsafe machinery.

- Reduces insurance rates because well maintained machinery increases safety.
- Reduces the time required to make machinery repairs - advance notice of machinery condition permits more efficient organization of the repair process.
- Increases the speed that machinery can be operated, if desirable.
- Increases the ease of operation of machinery.

1.2 Machine Learning

Machine learning algorithms search for patterns and regularities in any given data and have found wide usage across various application domains. They automatically learn from data by generalizing from examples. As more data becomes available more ambitious problems can be tackled. These algorithms are typically implemented in two phases. In the first phase, called training phase, data is gathered and provided to the algorithm, so it can learn patterns and create a model to classify data or predict data properties. In the second phase, called testing phase, new data is tested against the model that was built during the training phase, and the effectiveness of the model is revealed. Such two-phase learning algorithms are called supervised learning algorithms. There are also algorithms in which the testing phase is not used, and such algorithms are called unsupervised learning algorithms. These algorithms use unlabeled data to cluster the data in different classes. Machine learning algorithms can be used for classification or regression. In classification, the machine learning algorithm learns to classify the data in different classes while in regression it predicts a continuous variable by learning from the train data.

Under traditional programming models, programs and data are processed by the computer to produce a desired output, such as using programs to process data and produce a report (see Figure

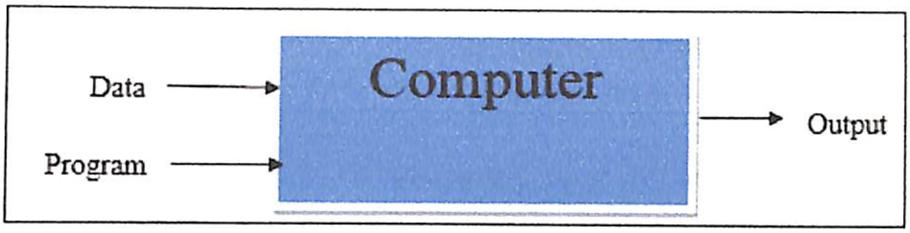


Figure 1 : Traditional Programming

When working with machine learning, the processing paradigm is altered dramatically. The data and the desired output are reverse-engineered by the computer to produce a new program, as shown in Figure 2.

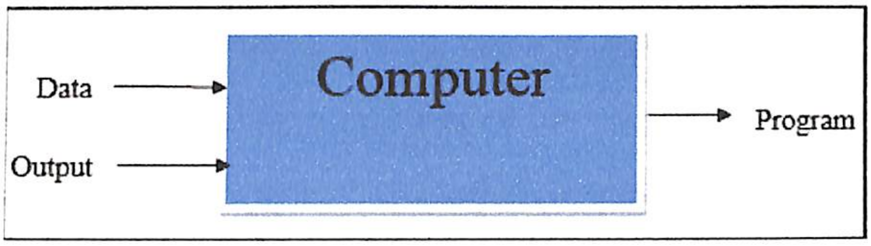


Figure 2 : Machine Learning

The power of this new program is that it can effectively "predict" the output, based on the supplied input data. The primary benefit of this approach is that the resulting "program" that is developed has been trained (via massive quantities of learning data) and finely tuned (via feedback data about the desired output) and is now capable of predicting the likelihood of a desired output based on the provided data. In a sense, it's equivalent to having the ability to create a goose that can lay golden eggs!

1.2.1 Supervised learning

Supervised learning is a type of machine learning algorithm that uses known datasets to create a model that can then make predictions. The known data sets are called and include input data elements along with known response values. From these training datasets, supervised learning algorithms attempt to build a new model that can make predictions based on new input values along with known outcomes.

Supervised learning can be separated into two general categories of algorithms:

Classification - These algorithms are used for predicting responses that can have just a few known values—such as married, single, or divorced—based on the other columns in the dataset.

Regression - These algorithms can predict one or more continuous variables, such as profit or loss, based on other columns in the dataset.

The formula for producing a supervised learning model is expressed in Figure.

Figure illustrates the general flow of creating new prediction models based on the use of supervised learning along with known input data elements and known outcomes to create an entirely new prediction model. A supervised learning algorithm analyzes the known inputs and known outcomes from training data. It then produces a prediction model based on applying algorithms that are capable of making inferences about the data.

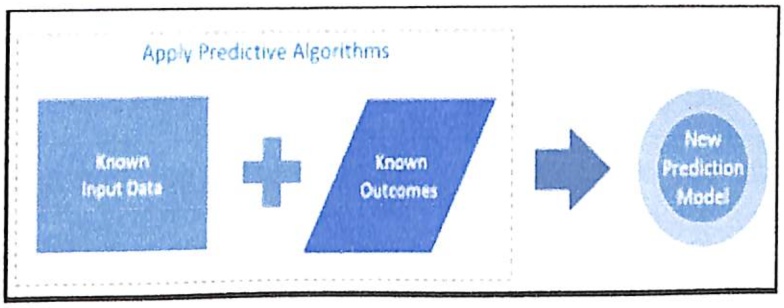


Figure 3 : Formula for supervised learning.

1.2.2 Unsupervised Learning

In the case of unsupervised machine learning, the task of making predictions becomes much harder. In this scenario, the machine learning algorithms are not provided with any kind of known data inputs or known outputs to generate a new predictive model.

In the case of unsupervised machine learning, the success of the new predictive model depends entirely on the ability to infer and identify patterns, structures, and relationships in the incoming data set.

The goal of inferring these patterns and relationships is that the objects within a group be similar to one another—and also different from other objects in other groups.

There are two basic approaches to unsupervised machine learning.

1.2.3 Why do we need machine learning?

Different temperatures, different operating environments—all those things should be factored in when looking at the data and coming up with accurate predictions of when to service and replace parts. Machine learning can provide this insight along with how long it takes to service the product, how much it costs to service it, and the combinations of factors that will lead to equipment failure such as the operating environment.

With a more holistic, micro view into the data coming from connected equipment, you can create a more comprehensive data story and uncover new ways to accurately and confidently lower the cost of producing and servicing products, generate new revenue opportunities, and sell more products and services at lower price points.

The failure of parts and components in large, complicated industrial machinery is an ever-present problem in the manufacturing world. What if a factory manager could be forewarned against failures of certain parts before they happened, minimizing down time? The U.S. Department of Energy cites surveys estimating the cost savings of a functioning predictive maintenance program to be 8-12 percent of total maintenance spending. In a world where margins are being squeezed from all sides, this can mean significant savings, especially when manufacturing on a large scale.

Imagine a machine that contains a hose of some sort, perhaps used to supply lubricant to a moving part in an industrial machine. Hoses commonly wear out, but it can be hard to predict when a hose failure will occur. Using a machine learning algorithm, one can monitor the operating condition of the hose in real-time to help establish normal operating parameters. For example, the plant manager could attach a vibrometer and thermometer to the hose and sample output from those sensors once per minute for several years and through several hose failures. Over time, the plant will build up a comprehensive dataset that will help predict hose failures. For the sake of simplicity, assume that hose failures are caused by a combination of heat and vibration. The large dataset will show which combinations of heat and vibration tend to coincide with or precede hose failures. Through data analysis the plant manager may find that when heat is more than three standard deviations above the mean hose temperature and when vibration is more than one standard deviation above the mean, hose failure is expected within one week in 95 percent of cases.

The next time the machinery exhibits those two conditions simultaneously, the machine learning algorithm may trigger an alert to the plant manager and also automatically place orders with suppliers for new hoses in anticipation of imminent failure. This way, the plant saves money either by mitigating the failure conditions (reducing vibration) or avoiding machine downtime by replacing the hose proactively.

1.3 Requirement Analysis

The Project requires some basic minimum resources so that it can function properly. This project has Functional as well as non-functional requirements that have to be full filled in order to make the application run properly. The program has to full fill the non-functional requirements to maintain the quality of its output and overall. All these requirements have been listed below briefly.

1.3.1 Requirements

The various Functional requirements of the system can be summarized as follows:

Table 1 : Basic functional Requirements

Supported Operating Systems	Windows 7 and above (x86 and x64)
Software Used	Microsoft Azure Ml: Microsoft Azure IOT Hub Stream Analytics Visual Studio C#
Supported architectures	32-bit (x86) 64-bit (x64)

1.4 Main Objectives

With recent advances in the Internet of Things, machine-to-machine communication, and connected systems, it is possible to stream sensor readings from machines so their health can be monitored in real time. Due to the high velocity and volume of data being streamed, it is not humanly possible to check all sensor values. But advances in predictive analytics have made it possible to evaluate the changes in the patterns of the sensor readings and identify the chances of machine failure significantly in advance of the event. It is also possible to identify the probability of other parts failing in due course by analyzing the historical patterns of failures, so that preventive maintenance can be planned. Capturing the changes in patterns in the sensor feeds in real time, and using that information to predict a possible failure and schedule a proactive maintenance strategy, is called predictive maintenance through the utilization of various nondestructive testing and measuring techniques, predictive maintenance determines equipment status before a breakdown occurs. To avoid unnecessary failure include timely, consistent equipment inspection and the aggressive use of nondestructive testing techniques such as vibration analysis, infrared testing, oil analysis and other techniques. Vibration analysis technique is used to avoid unnecessary failures.

1.5 Pert Chart Legend:

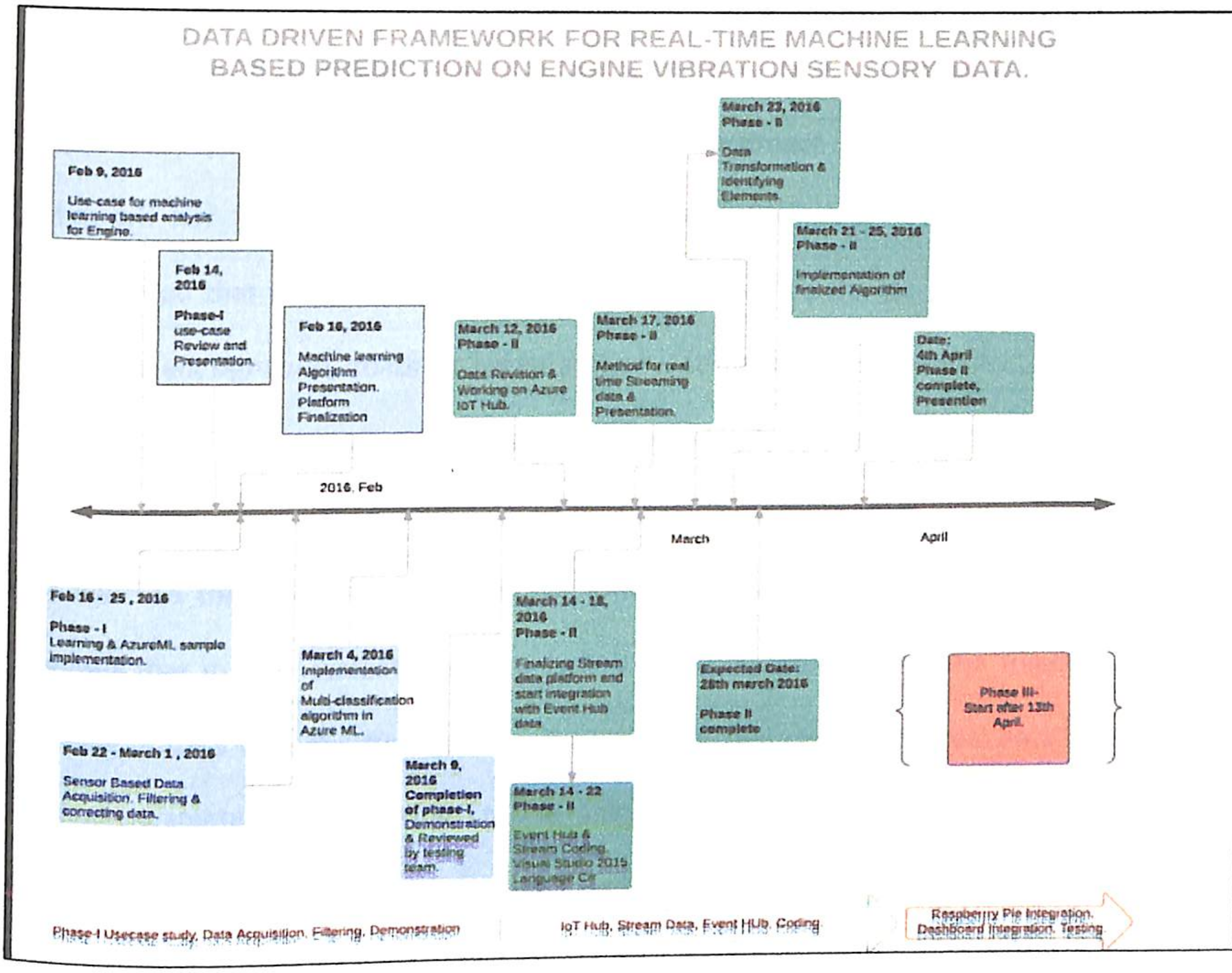


Figure 4 : Pert Chart

2. SYSTEM ANALYSIS

2.1 LITERATURE REVIEW

In [1] this paper the utilization of neural system approach for the condition observing framework has been accepted in an off-line proof of concept procedure. It has been demonstrated that the grouping of the machine framework parameters, on the premise of motion current signature, utilizing a neural system methodology is conceivable. Load inertia on an engine shaft has been characterized into various classifications on the premise of motion current signature. General right classification accuracy rate is observed to be 97.59%. Moreover, it is watched that the classification is not influenced by changing tuning parameters and proposes that motion current signature is free of tuning parameters. The freedom of the grouping process from the tuning parameters is a valuable result since real time changes in machine parameters are not coupled to the tuning parameter changes.

In [2] this paper grows prior work that built up the likelihood of the classification of machine system parameters on the premise of movement current mark, utilizing a neural system approach. The exactness of the system, to predict the adjustments in the estimation of the machine framework parameter, is an immediate function of the legitimacy of the simulated data, utilized for training the neural network. In this paper, the simulation model is approved against an on-line production machine. Different ways to deal with approve the recreation model are connected and a simulation model is produced. It has been demonstrated that the TuneLearn is equipped for mapping the motion current signature to the machine framework parameters. The sensitivity analysis of the TuneLearn brought about the 99.5% exactness of the sensitivity of the TuneLearn to the deviations in the estimations of the inertia, the friction torque and the gravitation torque.

In [3] this paper, to prepare the elements keeping in mind the end goal to figure the probability that every element value is ordinary statistical model is used. The yield of the statistical model is then sustained to a probabilistic classifier (regression) that is used for classification whether the case is normal or anomaly is there. Recognized irregularities are from that point sent on-board for further investigation utilizing case-based reasoning to predict the seriousness of a fault. Both the Gaussian blend model and the logistic relapse are thought to be based on-board the machine, however being utilized on-board the machine. In this manner, just restricted computational assets are required for the on-board preparing.

In [4], the paper introduces a way to deal with identifying anomaly groupings of individual oddities from, for example, the on-board anomaly indicator discovers same faulty gatherings of cases. Expecting that on-load up anomaly identifier may give back a substantial number of false positives after some time, it may regard recognize haphazardly happening abnormalities and a noteworthy increment in anomaly.

In [5], this paper researchers deal with intertwining the classification of the individual cases from the cases which are anomalous. A sliding window of the latest cases are separately named anomalous or not, and we can then distinguish whether an impossible number of cases are anomalous as a collective compared with a much bigger sliding window. At that point, the individual anomalous cases in the latest window are grouped utilizing the CBR classifier and from there on, a last characterization is surmised by grouping of the result of an individual classification.

2.2 Motivations

On Board Diagnostic (OBD) system provides information of the fuel economy, On-board diagnostics (OBD) is an automotive term referring to a vehicle's self-diagnostic and reporting capability. OBD systems give the vehicle owner or repair technician access to the status of the various vehicle subsystems. But for identifying the type of failure, vibration data is necessary. The vibration data is nothing but frequency and Amplitude.

The project aims on vibration reading collected analogous in form of Frequency (Hertz) and Amplitude .Analogous waves are further Fourier Transformed [Digital Form]. The major drawback of OBD system is, the data logger only provides information about fuel, car speed. Faulty Behavior of engine can only be classified based on its Amplitude (noise), distortion filtered and clean vibration data is accumulated for further processing.

2.3 Proposed system

Until now, machine maintenance has been either reactive (performed when failure occurs) or based on some heuristic (such as servicing the machine every "n" hours of continuous operation). Parts that fail earlier than expected can be very expensive and time consuming for businesses, because the necessary tools, components, and people may not be available for immediate response – resulting in machine downtime. But since such incidents are sporadic, it is not economically feasible for a company to stock all parts and tools and employ resources just to have them at the ready; this results in unnecessary costs. To counter this problem, companies perform scheduled maintenance using either statistical concepts like mean time to failure or other engineering practices based on historical experiences. However, this approach has its own challenge: parts can be replaced or serviced too frequently, which drives up costs unnecessarily. There are also instances where machines fail before the scheduled planned maintenance due to other reasons. Predictive maintenance is the complement of preventive maintenance. Through the utilization of various nondestructive testing and measuring techniques, predictive maintenance determines equipment status before a breakdown occurs. To avoid unnecessary failure include timely, consistent equipment inspection and the aggressive use of nondestructive testing techniques such as vibration analysis, infrared testing, oil analysis and other techniques. Vibration analysis technique is used to avoid unnecessary failures.

Classification algorithm is used for improving the Effectiveness & Accuracy of system Predictive maintenance decreases downtime by interpreting the condition data of an asset to potential failures in real-time.

3. System Design

UML Component Diagram

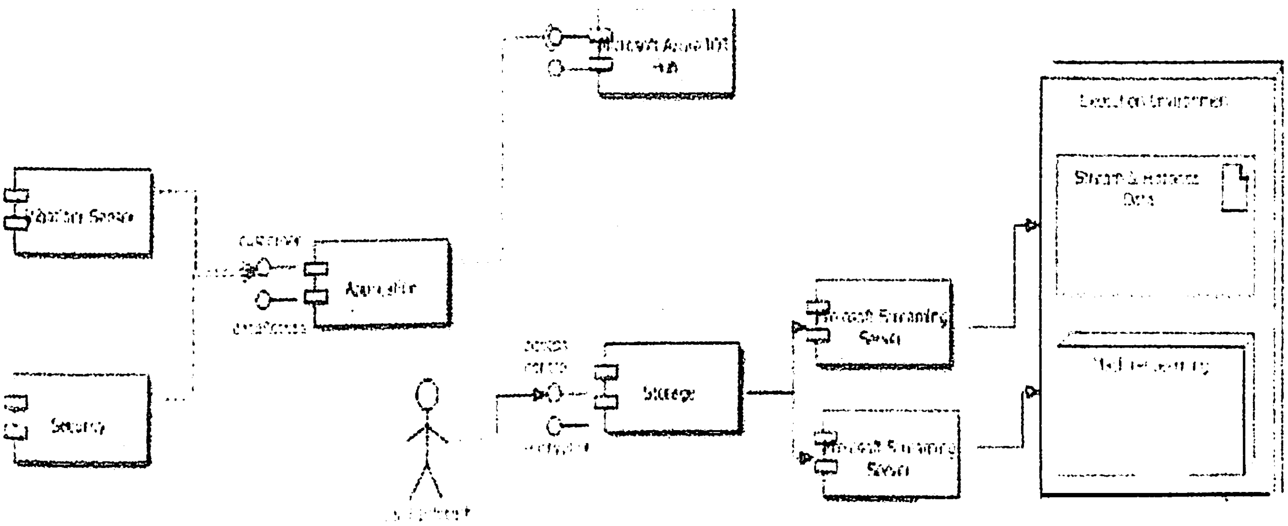


Figure 5: UML Diagram

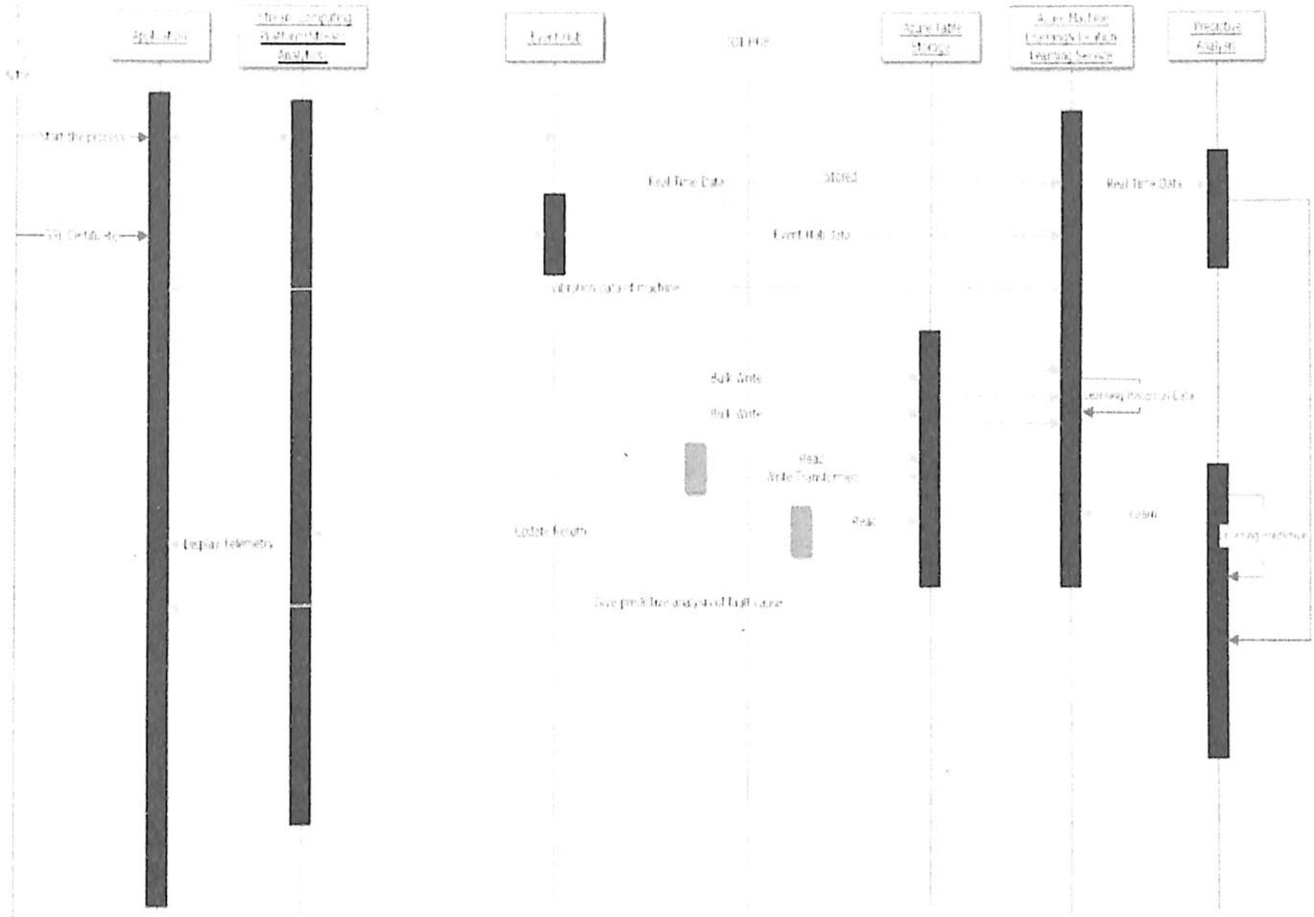


Figure 6: Sequence Diagram

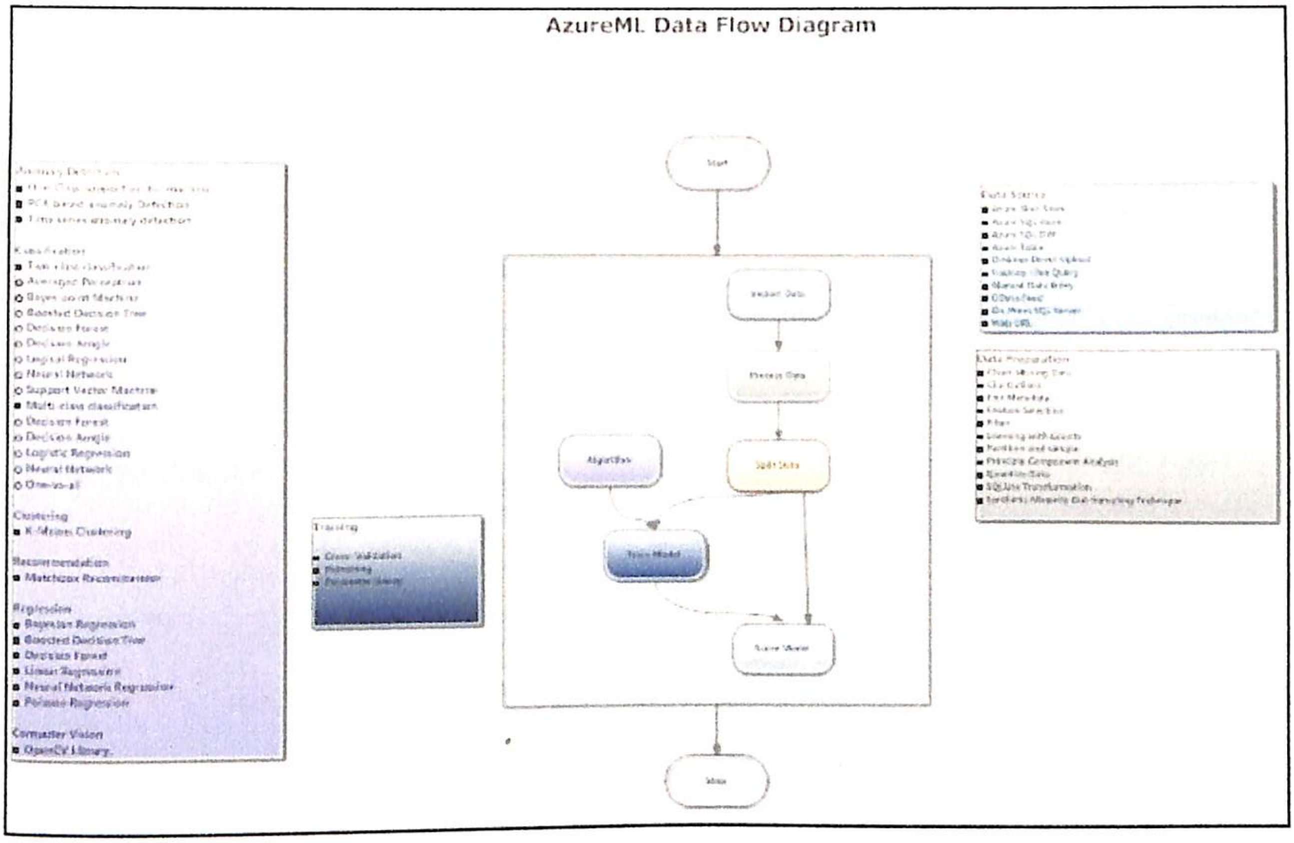


Figure 7 : DFD level 0

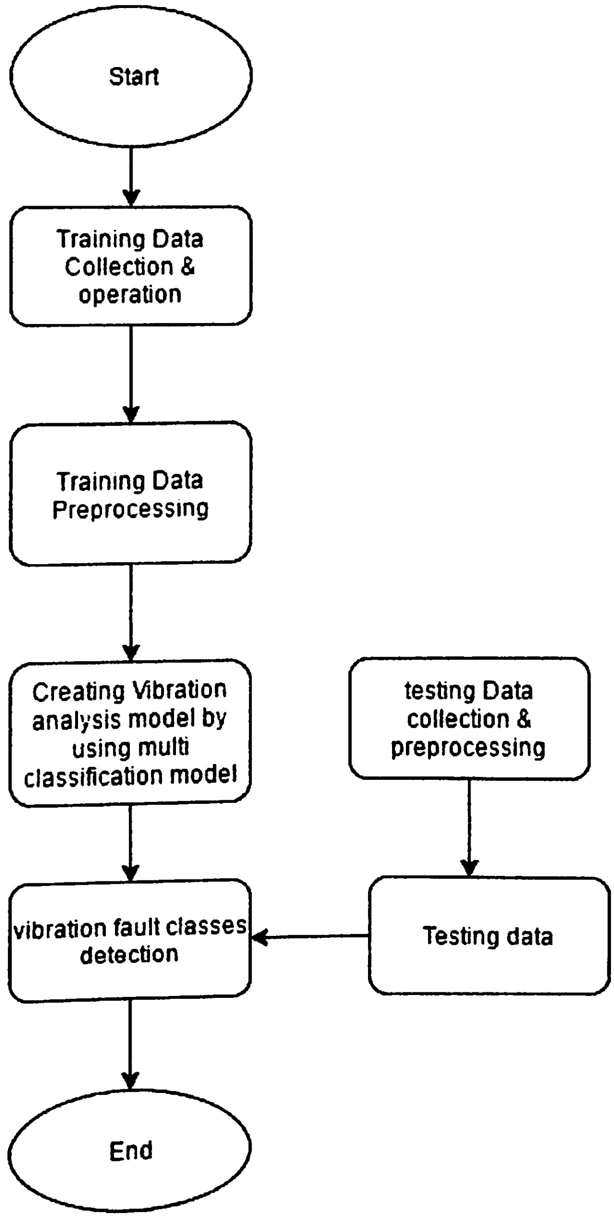


Figure 8 : DFD Level 1

4. PROJECT OVERVIEW

4.1 Technologies Used

4.1.1 Microsoft Azure ML

Azure Machine Learning is an effective cloud-based predictive analytic service that allows creating and deploying of predictive models. These Predictive models are created and deployed as analytic solutions. The basic process of creating Azure Machine Learning solutions is composed of a repeatable pattern of workflow steps that are designed to help create a new predictive analytics solution in less time.[6]

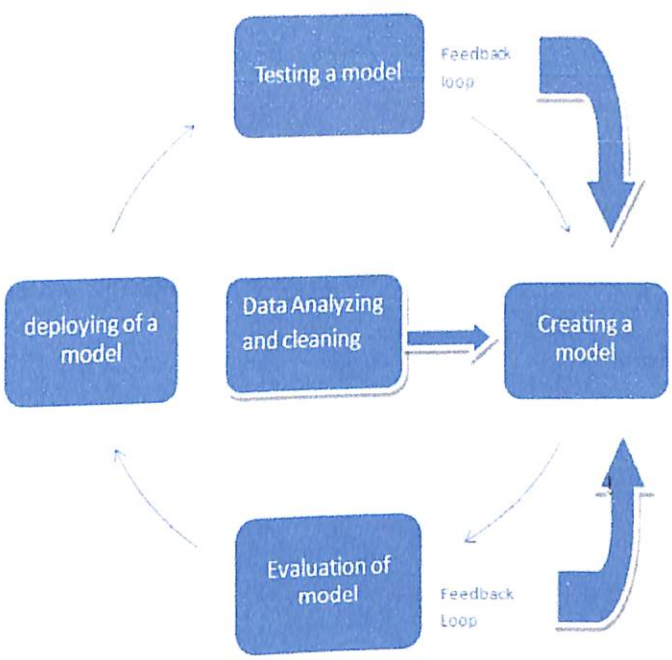


Figure 8 : Micosoft AzureML workflow

- Cleaning and analyzing of data: In this phase data is acquired, cleaned and transformed. Further data is split into training and testing datasets.
- Prediction model is created by applying various machine learning algorithms on datasets.

Model Evaluation: In advance both output and input values are known this is used to check the accuracy of predicted outcome given by new predictive model. Accuracy is measured in terms of confidence factor approaching the whole number one.

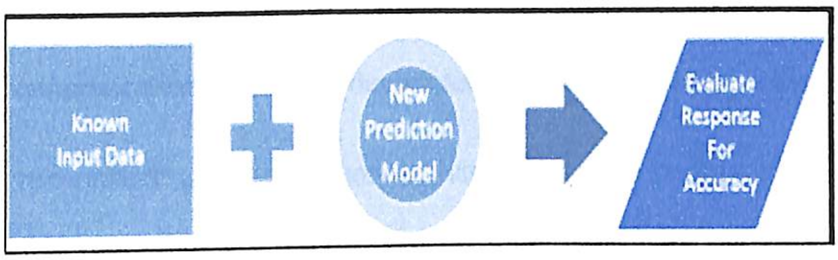


Figure 9 : Testing of Prediction Model

1) New predictive model is deployed as a cloud web service over the internet it can be accessible through any mobile client or web browser.

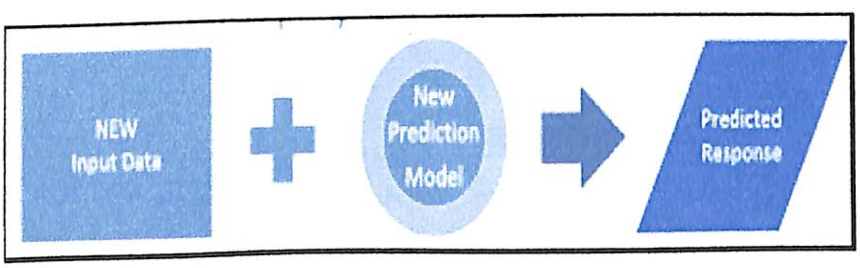


Figure 10 : Deploy New Model

2) Newly deployed predictive model web service is tested and implemented in any application. Feedback loops are added which will continuously improve the model by getting details of predicted outcome whether it is accurate or inaccurate. This will allow model to

continuously learn from its mistakes or inaccurate prediction it will never be repeated by model again.

4.1.2 Microsoft Azure IOT Hub

Azure IoT Hub is a fully managed service that enables reliable and secure bidirectional communications between millions of IoT devices and a solution back end. Azure IoT Hub:

- Provides reliable device-to-cloud and cloud-to-device messaging at scale.
- Enables secure communications using per-device security credentials and access control.
- Provides extensive monitoring for device connectivity and device identity management events.
- Includes device libraries for the most popular languages and platforms.

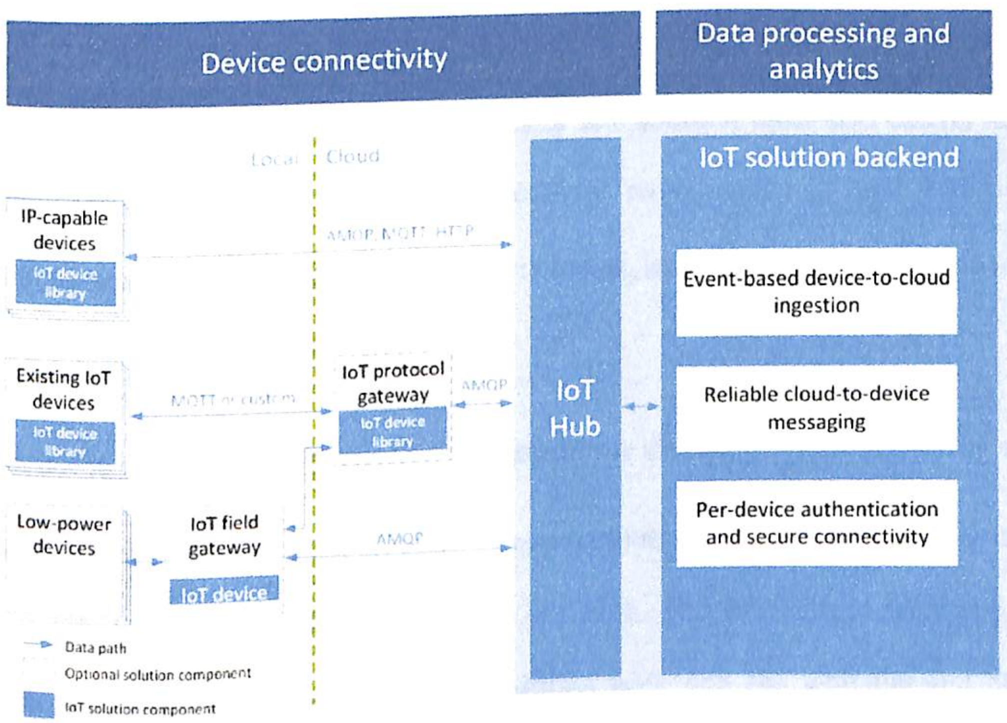


Figure 11 : IOT Solution Architecture

4.2.2.1 IoT device-connectivity challenges

IoT Hub and the device libraries help you to meet the challenges of how to reliably and securely connect devices to the solution back end. IoT devices:

- Are often embedded systems with no human operator.
- Can be in remote locations, where physical access is very expensive.
- May only be reachable through the solution back end.
- May have limited power and processing resources.
- May have intermittent, slow, or expensive network connectivity.
- May need to use proprietary, custom, or industry-specific application protocols.
- Can be created using a large set of popular hardware and software platforms.

In addition to the requirements above, any IoT solution must also deliver scale, security, and reliability. The resulting set of connectivity requirements is hard and time-consuming to implement when you use traditional technologies, such as web containers and messaging brokers.

4.2.2.2 Why use Azure IoT Hub

Azure IoT Hub addresses the device-connectivity challenges in the following ways:

- **Per-device authentication and secure connectivity**. You can provision each device with its own security key to enable it to connect to IoT Hub. The IoT Hub identity registry stores device identities and keys in a solution. A solution back end can whitelist and blacklist individual devices, which enables complete control over device access.

- **Monitoring of device connectivity operations.** Received detailed are operation logs about device identity management operations and device connectivity events. This enables your IoT solution to easily identify connectivity issues, such as devices that try to connect with wrong credentials, send messages too frequently, or reject all cloud-to-device messages.
- **An extensive set of device libraries.** Azure IoT device SDKs are available and supported for a variety of languages and platforms--C for many Linux distributions, Windows, and real-time operating systems. Azure IoT device SDKs also support managed languages, such as C#, Java, and JavaScript.
- **IoT protocols and extensibility.** If solution cannot use the device libraries, IoT Hub exposes a public protocol that enables devices to natively use MQTT v3.1.1, HTTP 1.1, or AMQP 1.0 protocols. You can also extend IoT Hub to provide support for custom protocol by customizing the the Azure IoT protocol gateway open source component. You can run the Azure IoT protocol gateway in the cloud or on-premises.
- **Scale.** Azure IoT Hub scales to millions of simultaneously connected devices and millions of events per second.

These benefits are generic to many communication patterns. IoT Hub currently enables you to implement the following specific communication patterns:

- **Event-based device-to-cloud ingestion.** IoT Hub can reliably receive millions of events per second from your devices. It can then process them on given hot path by using an event processor engine. It can also store them on your cold path for analysis. IoT Hub retains the event data for up to seven days to guarantee reliable processing and to absorb peaks in the load.

- **Reliable cloud-to-device messaging (or *commands*).** The solution back end can use IoT Hub to send messages with an at-least-once delivery guarantee to individual devices. Each message has an individual time-to-live setting, and the back end can request both delivery and expiration receipts. This ensures full visibility into the life cycle of a cloud-to-device message.Then it can implement business logic that includes operations that run on devices.

4.2.2.2 How does IoT Hub work?

Azure IoT Hub implements the service-assisted communication pattern to mediate the interactions between your devices and your solution back end. The goal of service-assisted communication is to establish trustworthy, bidirectional communication paths between a control system, such as IoT Hub, and special-purpose devices that are deployed in untrusted physical space. The pattern establishes the following principles:

- Security takes precedence over all other capabilities.
- Devices do not accept unsolicited network information. A device establishes all connections and routes in an outbound-only fashion. For a device to receive a command from the back end, the device must regularly initiate a connection to check for any pending commands to process.
- Devices should only connect to or establish routes to well-known services they are peered with, such as IoT Hub.
- The communication path between device and service or between device and gateway is secured at the application protocol layer.

- System-level authorization and authentication are based on per-device identities. They make access credentials and permissions nearly instantly revocable.
- Bidirectional communication for devices that connect sporadically due to power or connectivity concerns is facilitated by holding commands and device notifications until a device connects to receive them. IoT Hub maintains device specific queues for the commands it sends.
- Application payload data is secured separately for protected transit through gateways to a particular service.

Table 2 : Benefits of using MS Azure IOT Hub

AREA	IOT Hub	Event Hubs
Communication patterns	Enables device-to-cloud and cloud-to-device messaging	Only enables event in ingress (usually considered for device-to-cloud scenarios).
Device protocol support	Supports AMQP, AMQP over webSockets MQTT, and HTTP/1. Additionally IoT Hub works with the Azure IoT Protocol Gateway, a customizable protocol gateway	Supports AMQP, AMQP over WebSockets, and HTTP/1

	implementation to support custom protocols.	
Security	Provides per-device identity and revocable access control. See theSecurity section of the IoT Hub developer guide	Provides Event Hubs-wideshared access policies, with limited revocation support through publisher's policies. IoT solutions are often required to implement a custom solution to support per-device credentials and anti-spoofing measures.
Operations monitoring	Enables IoT solutions to subscribe to a rich set of device identity management and connectivity events such as individual device authentication errors, throttling, and bad format exceptions. These events enable you to quickly identify connectivity	Exposes only aggregate metrics.

	problems at the individual device level.	
Scale	Is optimized to support millions of simultaneously connected devices	Can support a more limited number of simultaneous connections--up to 5,000 AMQP connections, as per Azure Service Bus quotas. On the other hand, Event Hubs enables you to specify the partition for each message sent
Device SDKs	Provides device SDKs for a large variety of platforms and languages	Is supported on .NET, and C. Also provides AMQP and HTTP send interfaces.

In summary, even if the only use case is device-to-cloud telemetry ingress, IoT Hub provides a service that is specifically designed for IoT device connectivity. It will continue to expand the value propositions for these scenarios with IoT-specific features. Event Hubs is designed for event ingress at a massive scale, both in the context of inter-datacenter and intra-datacenter scenarios. It is not uncommon to use both IoT Hub and Event Hubs in the same solution--where IoT Hub handles the device-to-cloud communication and Event Hubs handles later-stage event ingress into real-time processing engines.

4.2.2.4 IoT solution architecture

The following diagram shows a typical IoT solution architecture. Note that it does not include the names of any specific Azure services, but describes the key elements in a generic IoT solution architecture. In this architecture, IoT devices collect data which they send to a cloud gateway. The cloud gateway makes the data available for processing by other back-end services from where data is delivered to other line-of-business applications or to human operators through a dashboard or other presentation device.

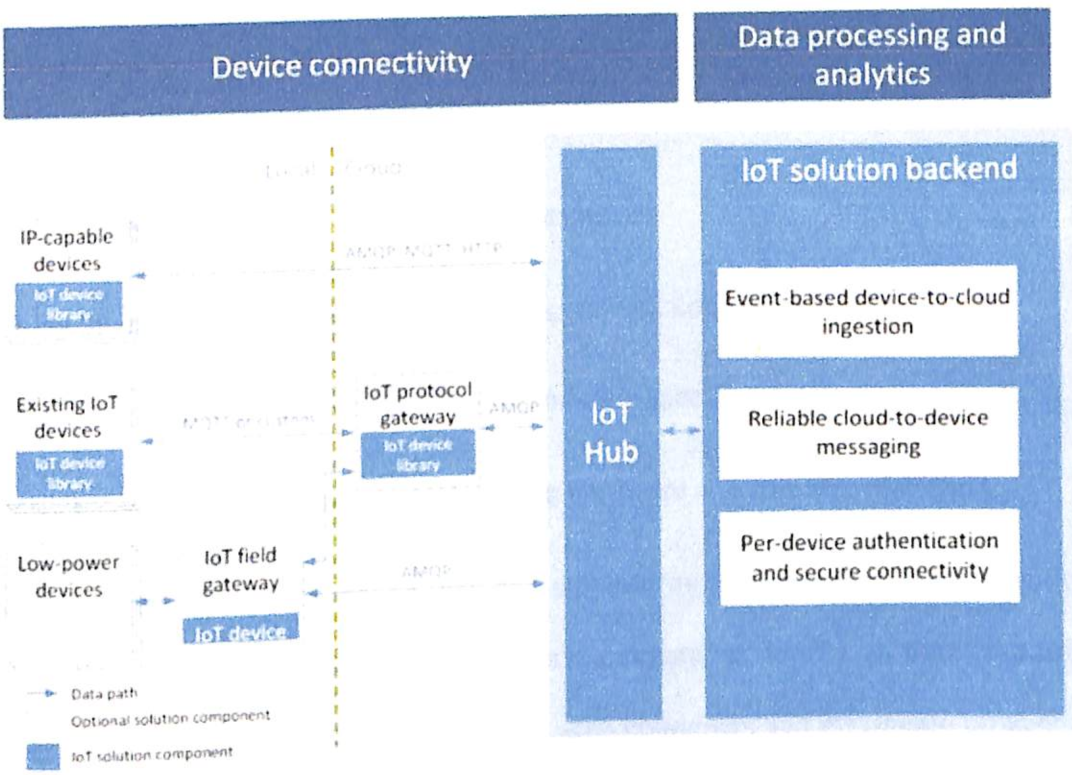


Figure 12: IOT Architrcture

4.2.2.5 Device connectivity

In this IoT solution architecture, devices send telemetry, such as temperature readings, to a cloud endpoint for storage and processing. Devices can also receive and respond to cloud-to-device commands by reading messages from a cloud endpoint. For example, a device might retrieve a command that instructs it to change the frequency at which it samples data.

One of the biggest challenges facing IoT projects is how to reliably and securely connect devices to the solution back end. IoT devices have different characteristics as compared to other clients such as browsers and mobile apps. IoT devices:

- Are often embedded systems with no human operator.
- Can be located in remote locations, where physical access is very expensive.
- May only be reachable through the solution back end. There is no other way to interact with the device.
- May have limited power and processing resources.
- May have intermittent, slow, or expensive network connectivity.
- May need to use proprietary, custom, or industry specific application protocols.
- Can be created using a large set of popular hardware and software platforms.

In addition to the requirements above, any IoT solution must also deliver scale, security, and reliability. The resulting set of connectivity requirements is hard and time-consuming to implement using traditional technologies such as web containers and messaging brokers. Azure IoT Hub and the IoT Device SDKs make it easier to implement solutions that meet these requirements.

A device can communicate directly with a cloud gateway endpoint, or if the device cannot use any of the communications protocols that the cloud gateway supports, it can connect through an intermediate gateway, such as the IoT Hub protocol gateway, that performs protocol translation.

4.4.2.6 Data processing and analytics

In the cloud, an IoT solution back end is where most of the data processing in the solution occurs, in particular filtering and aggregating telemetry and routing it to other services. The IoT solution back end:

- Receives telemetry at scale from your devices and determines how to process and store that data.
- May enable you to send commands from the cloud to specific device.
- Provides device registration capabilities that enable you to provision devices and to control which devices are permitted to connect to your infrastructure.
- Enables you to track the state of your devices and monitor their activities.

IoT solutions can include automatic feedback loops. For example, an analytics module in the back end can identify from telemetry that the temperature of a specific device is above normal operating levels and then send a command to the device, enabling it to take corrective action.

4.4.2.7 Presentation and business connectivity

The presentation and business connectivity layer allows end users to interact with the IoT solution and the devices. It enables users to view and analyze the data collected from their devices. These views can take the form of dashboards or BI reports. For example, an operator can check on the status of particular shipping trucks and see any alerts raised by the system. This layer also allows integration of the IoT solution back end with existing line-of-business applications to tie into enterprise business processes or workflows.

4.1.3 Azure Table Storage:

The Azure Table storage service stores large amounts of structured data. The service is a NoSQL data store which accepts authenticated calls from inside and outside the Azure cloud. Azure tables are ideal for storing structured, non-relational data. Common uses of the Table service include:

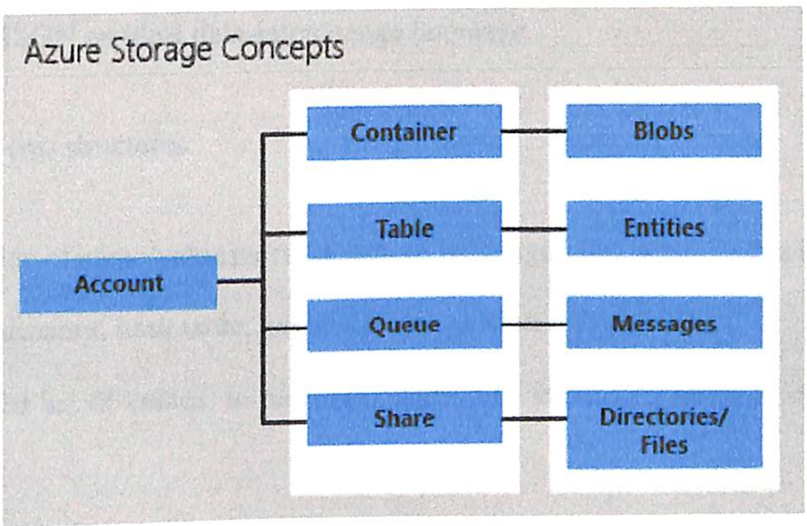


Figure 13 : Azure Storage Concepts

Storing TBs of structured data capable of serving web scale applications

- Storing datasets that don't require complex joins, foreign keys, or stored procedures and can be denormalized for fast access
- Quickly querying data using a clustered index

- Accessing data using the OData protocol and LINQ queries with WCF Data Service .NET Libraries

4.1.4 JSON:

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

JSON is built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an *object*, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an *array*, vector, list, or sequence.

These are universal data structures. Virtually all modern programming languages support them in one form or another. It makes sense that a data format that is interchangeable with programming languages also be based on these structures.

In JSON, they take on these forms:

An *object* is an unordered set of name/value pairs. An object begins with { (left brace) and ends with } (right brace). Each name is followed by : (colon) and the name/value pairs are separated by , (comma).

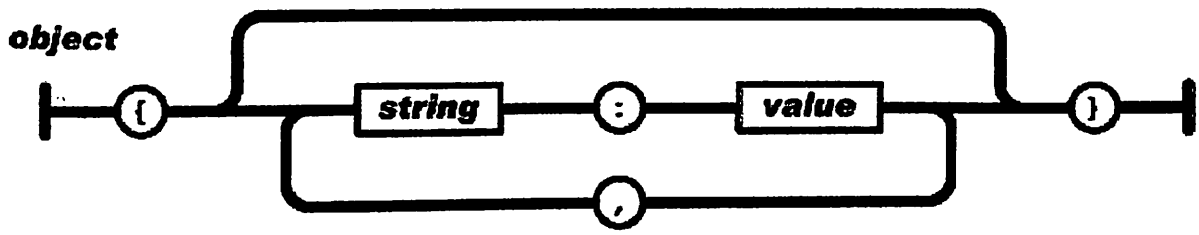


Figure 14:JSON

An *array* is an ordered collection of values. An array begins with [(left bracket) and ends with] (right bracket). Values are separated by, (comma).

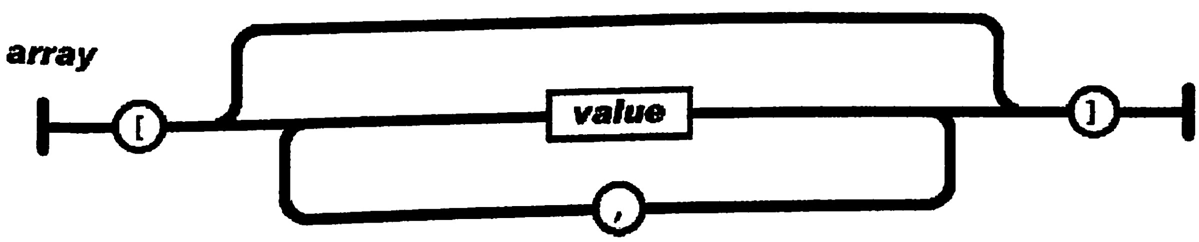


Figure 15 : JSON Structure

A *value* can be a *string* in double quotes, or a *number*, or true or false or null, or an *object* or an *array*. These structures can be nested.

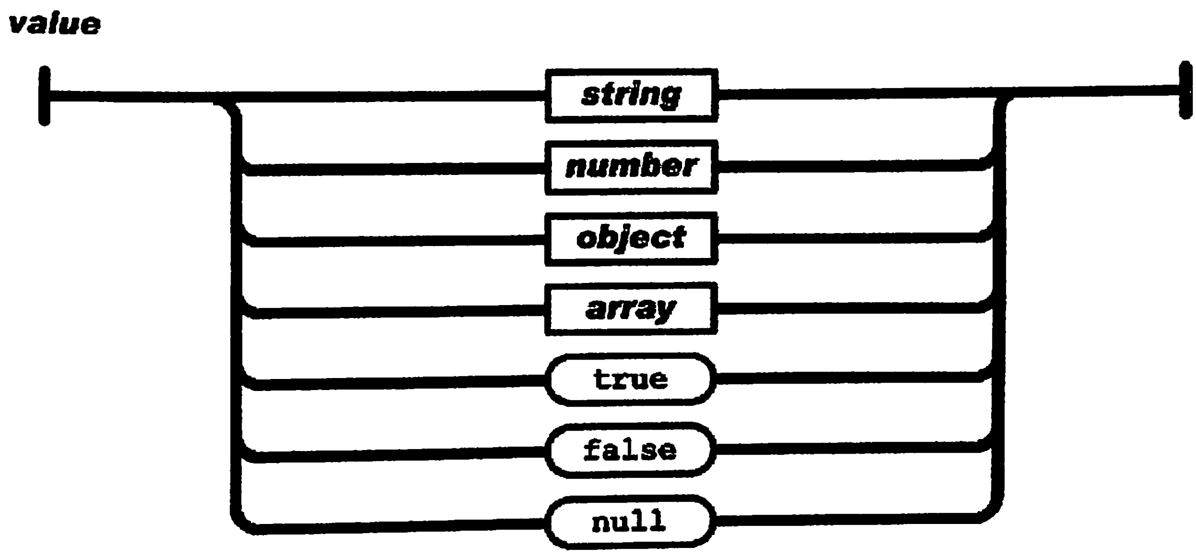


Figure 16 JSON Values

A *string* is a sequence of zero or more Unicode characters, wrapped in double quotes, using backslash escapes. A character is represented as a single character string. A string is very much like a C or Java string.

A *number* is very much like a C or Java number, except that the octal and hexadecimal formats are not used.

4.1.5 Microsoft Azure Cloud

Cloud Services is an example of Platform-as-a-Service (PaaS). This technology is meant to support applications that are scalable, reliable, and low-cost to control. rather like an App Service is hosted on VMs, thus too are Cloud Services, however, you've got a lot of management over the VMs.More control also means less ease of use; unless you need the additional control options, it's typically quicker and easier to get a web application up and running in Web Apps in App Service compared to Cloud Services.

The technology provides two slightly different VM options: instances of *web roles* run a variant of Windows Server with IIS, while instances of *worker roles* run the same Windows Server variant without IIS. A Cloud Services application relies on some combination of these two options:

Any combination of these two slightly different VM hosting options are available in a cloud service:

- Web-role

  Runs Windows Server with your web app automatically deployed to IIS.

- Worker-role

  Runs Windows Server without IIS

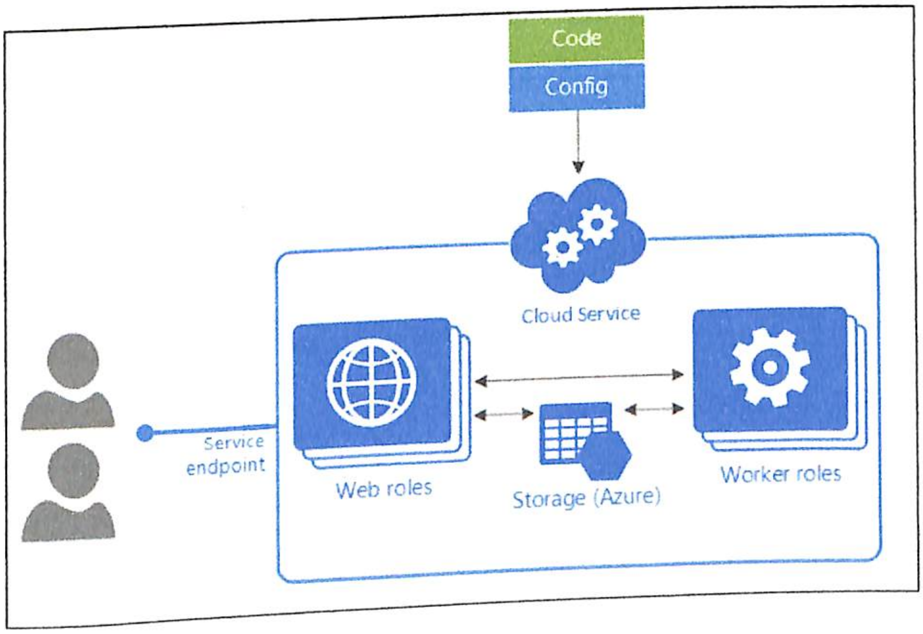


Figure 17: Azure Cloud

5. Implementation

5.1. Multiclass Decision Forest:

The decision forest algorithm is an ensemble learning method for classification. The algorithm works by building multiple decision trees and then *voting* on the most popular output class. Voting is a form of aggregation, in which each tree in a classification decision forest outputs a non-normalized frequency histogram of labels. The aggregation process sums these histograms and normalizes the result to get the "probabilities" for each label. The trees that have high prediction confidence will have a greater weight in the final decision of the ensemble.

Decision trees in general are non-parametric models, meaning they support data with varied distributions. In each tree, a sequence of simple tests is run for each class, increasing the levels of a tree structure until a leaf node (decision) is reached.

- Decision trees have many advantages:
- They can represent non-linear decision boundaries.
- They are efficient in computation and memory usage during training and prediction.
- They perform integrated feature selection and classification.
- They are resilient in the presence of noisy features.

5.2 Data Set Description

Table 3: Dataset Description

Data Title	Diagnosis of faults in electro-mechanical devices from vibration measurements
Attribute Name	**Description**
Class	The predicted Class
Component Number	Identification of Compoent of the pump
Support Number	Support in machine where measure was taken
Frequency	Frequency of the measure
Measure	Current Measure
Earlier Measure	Earlier current measure
Dir	Direction value
Omega	RPM of machine

Table 4: Dataset

Dir value	**filter present**	**type of the measure**	**direction**
Vo	no filter	Velocity	horizontal
Va	no filter	Velocity	axial
Vv	no filter	Velocity	vertical
Ao	no filter	Amplitude	horizontal
Aa	no filter	Amplitude	axial
Av	no filter	Amplitude	vertical
Io	Filter	Velocity	horizontal
Ia	Filter	Velocity	axial
Iv	Filter	Velocity	vertical

Predicted fault classes description:

Class	Type
1	Problems in the joint
2	Faulty bearing
3	Mechanical loosening
4	Basement distortion
5	Unbalance
6	Normal operating condition

Table 5: Predicted Classes Description

A vibration telematics simulator is included as part of this solution. It emits diagnostic information and signals corresponding to the state of the vehicle and driving pattern at a given point in time.

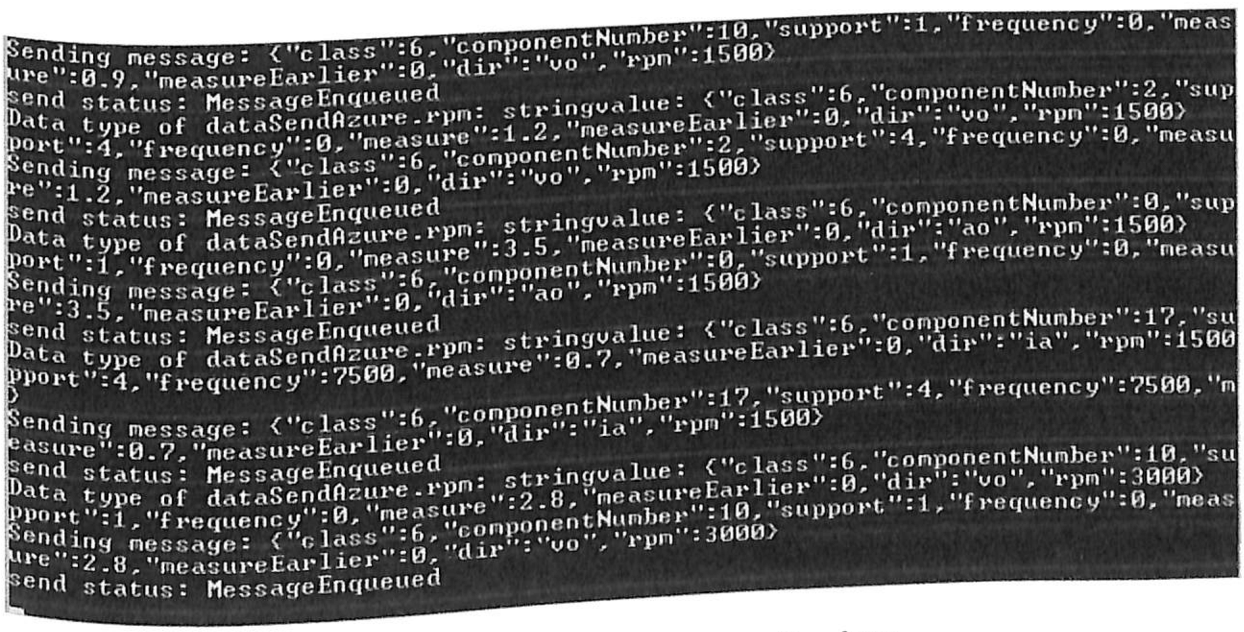
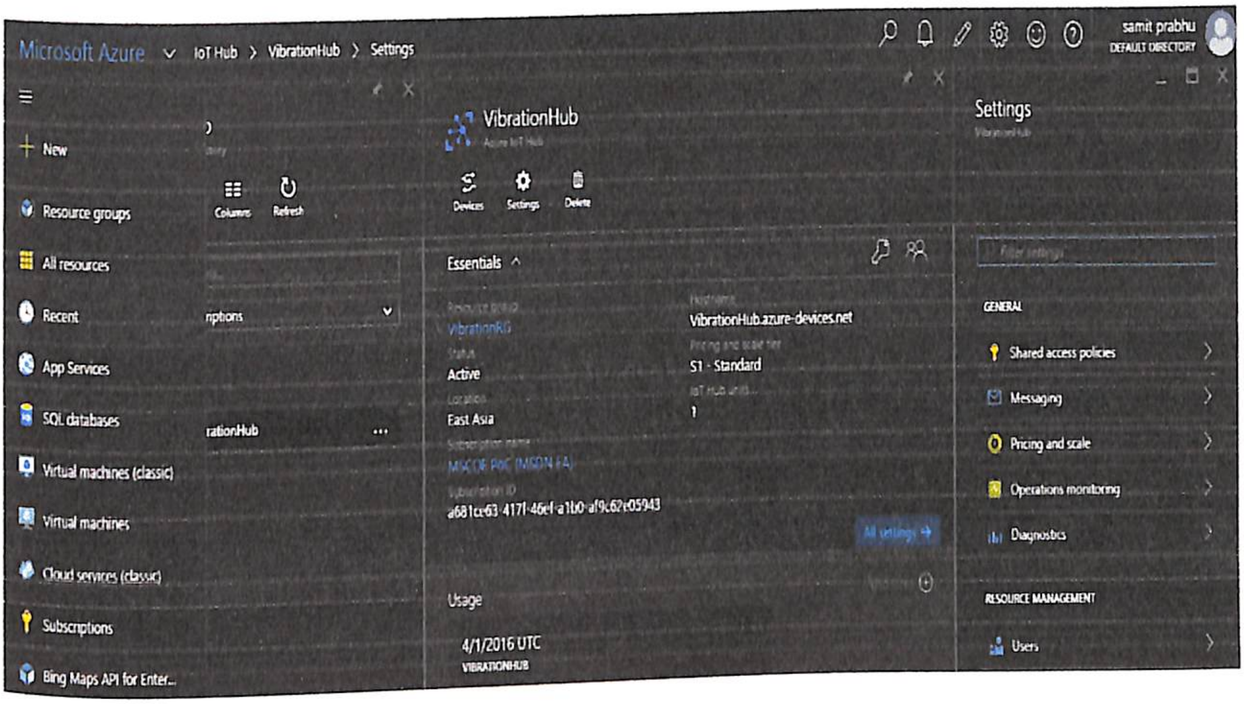


Figure 18 : Vibration Telematics Simulator

This data is JSON type and its description is explained above

5.3IOT HUB



1. In Visual Studio, add a new Visual C# Windows Classic Desktop project to the current solution using the Console Application project template. Name the projectCreateDeviceIdentity.

2. In Solution Explorer, right-click the CreateDeviceIdentity project, and then click Manage NuGet Packages.

3. In the NuGet Package Manager window, select Browse, search for microsoft.azure.devices, click Install to install the Microsoft.Azure.Devices package, and accept the terms of use.

4. This downloads, installs, and adds a reference to the Microsoft Azure IoT Service SDK NuGet package.

5. Add the following statements using at the top of the Program.cs file:

using Microsoft.Azure.Devices;

using Microsoft.Azure.Devices.Common.Exceptions;

1. Add the following fields to the Program class, replacing the placeholder value with the connection string for the IoT hub you created in the previous section:

```
static RegistryManager registryManager;
static string connectionString = "{iothub connection string}";
```

2. Add the following method to the Program class:

```
private async static Task AddDeviceAsync()
{
string deviceId = "tDevice";
Device device;
try
{
device = await registryManager.AddDeviceAsync(new Device(deviceId));
}
catch (DeviceAlreadyExistsException)
{
device = await registryManager.GetDeviceAsync(deviceId);
}
Console.WriteLine("Generated        device key:    {0}",
device.Authentication.SymmetricKey.PrimaryKey);
}
```

This method creates a new device identity with ID Device (if that device ID already exists in the registry, the code simply retrieves the existing device information). The app then displays the primary key for that identity. You will use this key in the simulated device to connect to your IoT hub.

3. Finally, add the following lines to the Main method:

registryManager = RegistryManager.CreateFromConnectionString(connectionString);

AddDeviceAsync().Wait();

Console.ReadLine();Run this application, and make a note of the device key.

5.4 Stream Analytics(Real Time Analysis)

The events generated by the Vibration Telematics Simulator are published to the Event Hub using the Event Hub SDK. The Stream Analytics job ingests these events from the Event Hub and processes the data in real-time to analyze the machine health.

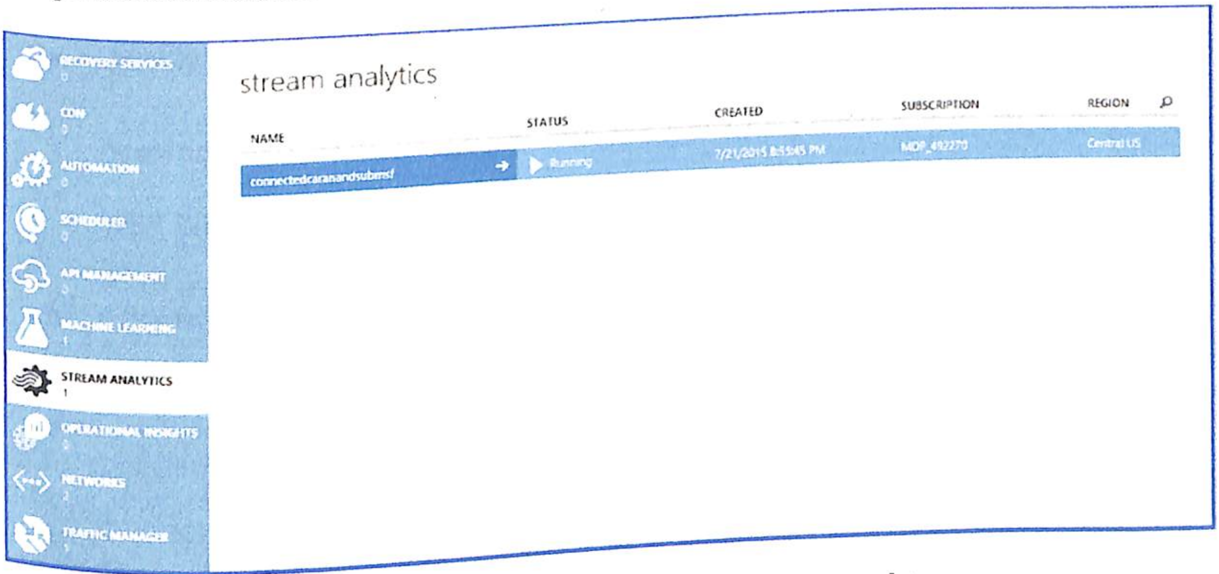


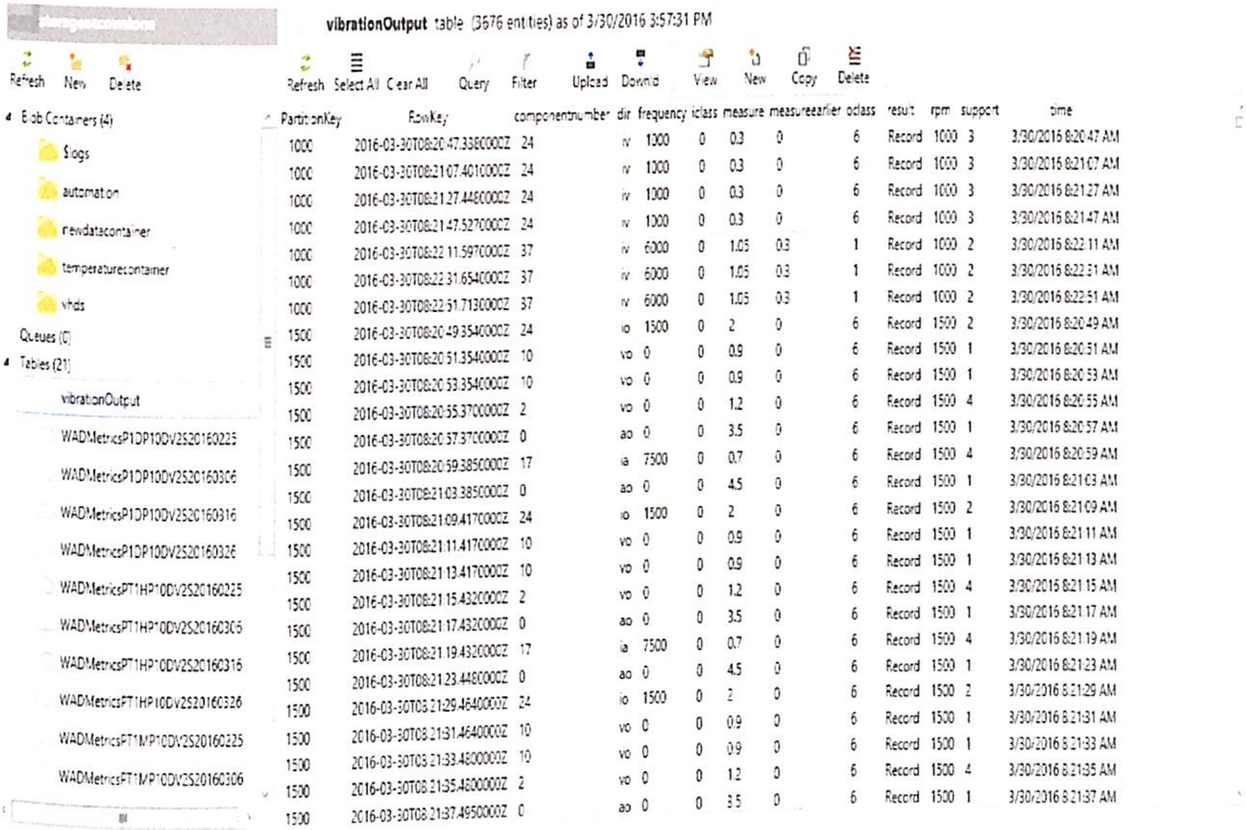
Figure 19: Stream analytics job processing data

The stream analytics job ingests data from the Event Hub, performs a join with the reference data to the corresponding model and also persists them into Azure blob storage for rich batch analytics. The below stream analytics query is used to persist the data into Azure table storage.

5.5 Azure Table Storage

The Azure Table storage service stores large amounts of structured data. The service is a NoSQL data store which accepts authenticated calls from inside and outside the Azure cloud. Azure tables are ideal for storing structured, non-relational data. Common uses of the Table service include:

- Storing TBs of structured data capable of serving web scale applications
- Storing datasets that don't require complex joins, foreign keys, or stored procedures and can be denormalized for fast access
- Quickly querying data using a clustered index
- Accessing data using the OData protocol and LINQ queries with WCF Data Service .NET Libraries
- After stream analytics process is done the data is stored in Azure Table Storage.



PartitionKey	RowKey	componentnumber	dir	frequency	iclass	measure	measureearlier	oclass	result	rpm	support	time
1000	2016-03-30T08:20:47.3380000Z	24	iv	1000	0	0.3	0	6	Record	1000	3	3/30/2016 8:20:47 AM
1000	2016-03-30T08:21:07.4010000Z	24	iv	1000	0	0.3	0	6	Record	1000	3	3/30/2016 8:21:07 AM
1000	2016-03-30T08:21:27.4480000Z	24	iv	1000	0	0.3	0	6	Record	1000	3	3/30/2016 8:21:27 AM
1000	2016-03-30T08:21:47.5270000Z	24	iv	1000	0	0.3	0	6	Record	1000	3	3/30/2016 8:21:47 AM
1000	2016-03-30T08:22:11.5970000Z	37	iv	6000	0	1.05	0.3	1	Record	1000	2	3/30/2016 8:22:11 AM
1000	2016-03-30T08:22:31.6540000Z	37	iv	6000	0	1.05	0.3	1	Record	1000	2	3/30/2016 8:22:31 AM
1000	2016-03-30T08:22:51.7130000Z	37	iv	6000	0	1.05	0.3	1	Record	1000	2	3/30/2016 8:22:51 AM
1500	2016-03-30T08:20:49.3540000Z	24	io	1500	0	2	0	6	Record	1500	2	3/30/2016 8:20:49 AM
1500	2016-03-30T08:20:51.3540000Z	10	vo	0	0	0.9	0	6	Record	1500	1	3/30/2016 8:20:51 AM
1500	2016-03-30T08:20:53.3540000Z	10	vo	0	0	0.9	0	6	Record	1500	1	3/30/2016 8:20:53 AM
1500	2016-03-30T08:20:55.3700000Z	2	vo	0	0	1.2	0	6	Record	1500	4	3/30/2016 8:20:55 AM
1500	2016-03-30T08:20:57.3700000Z	0	ao	0	0	3.5	0	6	Record	1500	1	3/30/2016 8:20:57 AM
1500	2016-03-30T08:20:59.3850000Z	17	ia	7500	0	0.7	0	6	Record	1500	4	3/30/2016 8:20:59 AM
1500	2016-03-30T08:21:03.3850000Z	0	ao	0	0	4.5	0	6	Record	1500	1	3/30/2016 8:21:03 AM
1500	2016-03-30T08:21:09.4170000Z	24	io	1500	0	2	0	6	Record	1500	2	3/30/2016 8:21:09 AM
1500	2016-03-30T08:21:11.4170000Z	10	vo	0	0	0.9	0	6	Record	1500	1	3/30/2016 8:21:11 AM
1500	2016-03-30T08:21:13.4170000Z	10	vo	0	0	0.9	0	6	Record	1500	1	3/30/2016 8:21:13 AM
1500	2016-03-30T08:21:15.4320000Z	2	vo	0	0	1.2	0	6	Record	1500	4	3/30/2015 8:21:15 AM
1500	2016-03-30T08:21:17.4320000Z	0	ao	0	0	3.5	0	6	Record	1500	1	3/30/2016 8:21:17 AM
1500	2016-03-30T08:21:19.4320000Z	17	ia	7500	0	0.7	0	6	Record	1500	4	3/30/2016 8:21:19 AM
1500	2016-03-30T08:21:23.4480000Z	0	ao	0	0	4.5	0	6	Record	1500	1	3/30/2016 8:21:23 AM
1500	2016-03-30T08:21:29.4640000Z	24	io	1500	0	2	0	6	Record	1500	2	3/30/2016 8:21:29 AM
1500	2016-03-30T08:21:31.4640000Z	10	vo	0	0	0.9	0	6	Record	1500	1	3/30/2016 8:21:31 AM
1500	2016-03-30T08:21:33.4800000Z	10	vo	0	0	0.9	0	6	Record	1500	1	3/30/2016 8:21:33 AM
1500	2016-03-30T08:21:35.4800000Z	2	vo	0	0	1.2	0	6	Record	1500	4	3/30/2016 8:21:35 AM
1500	2016-03-30T08:21:37.4950000Z	0	ao	0	0	3.5	0	6	Record	1500	1	3/30/2016 8:21:37 AM

Figure 20: Azure Table Storage

5.6 Azure Cloud Storage

Cloud Services is an example of Platform-as-a-Service (PaaS). This technology is designed to support applications that are scalable, reliable, and cheap to operate. Just like an App Service is hosted on VMs, so too are Cloud Services, however, you have more control over the VMs. Web role is used for hosting web app.

- Web role

  Runs Windows Server with your web app automatically deployed to IIS

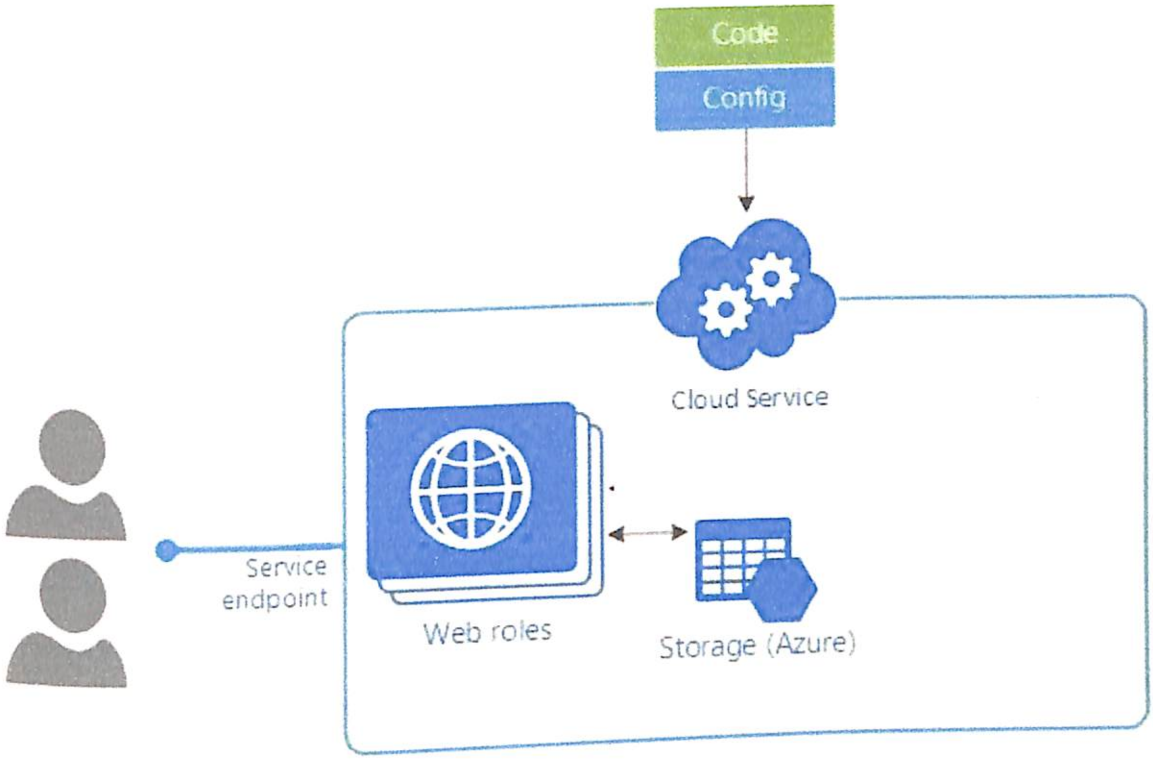


Figure 21: Azure Cloud Storage

6. Limitations& Future Enhancements

6.1 Limitations

The down side of using a predictive maintenance approach is its initial costs. The up-front costs of starting this type of program can be expensive. Much of the equipment requires expenditures.. And, beginning a predictive maintenance program requires an understanding of the facility's predictive maintenance needs and the approaches which need to be undertaken. It is also essential to have a firm commitment, by management and all facility staff and organizations, to make it work.

There are two related aspects that make building models for predicting rare events especially difficult. First, due to the very fact that they are rare, the training data does not contain many examples of these events. Given that such events are often complex (for example, the sensor readings at the time of failure may not point to the cause of failure, since the failure could have been the result of a buildup of several other events already reported as past sensor readings), it is often not easy to generalize from the few available examples. Second, depending on how predictive algorithms are configured and what underlying cost functions are, classifiers or models might completely ignore the rare event. Therefore, special attention should be given to the choice of performance metric being used in general and, more important, for cases with unbalanced classes (where one class of responses is disproportionately larger than the opposite class).In addition to this, there are several other complexities: not all parts of a machine fail with the same frequency, the type of failure can vary, and more. Because sensor data is streamed in real time, it can have missing values or garbage values as well. The frequency at which each sensor transmits data can also vary. Hence data preparation and modeling techniques are needed to help prepare the data and build predictive models to address the nature of the problem and capture the necessary information to effectively predict the failure or breakdown of the machine

6.2 Future Enhancement:

1. More Sophisticated Vehicle Connectivity

As vehicles get 'smarter,' maintenance and repair diagnostics will be sent directly from the vehicle and, based on the diagnostic results, the vehicle's interface screen will route the driver to the appropriate shop.

Future enhancements can be:

- Sophisticated vehicle connectivity.
- Improved vehicle maintenance analytics.
- Continued parts shortages.
- Ongoing technician shortages and resulting higher labor rates.

2. Maintenance Data Management

The growth in amount of vehicle data and information will be used for analytics that will give result in best-in-class recommendations

3. Expansion of Preventive Maintenance Reminders

There will be an increase in onboard reminders for preventive maintenance beyond traditional oil changes and tire pressure. Sensors in filters and fluids and other maintenance items will trigger dashboard reminders to alert the driver of due or overdue maintenance. This growing technology should help reduce related component failures and roadside breakdowns.

4. Greater Tool Standardization

The continuing move by OEMs to develop global vehicle platforms will help reduce the need for unique tools required by automotive technicians. A single tool will be able to function across multiple vehicle models. This will help control repair providers' tool costs and simplify the tool selection process.

5. Improved Communication with Drivers

More automation and interactions with vendors will provide improved communications for drivers (such as text messaging repair status updates and apps to schedule service appointments, and communicate local pricing promotions) as well as improved data for FMCs to monitor and rate a shop's performance.

7. Conclusionand Results

Machine learning is used by various manufacturing industries for developing various applications. By using machine learning in real time data stream decision making becomes more accurate. In this thesis future prediction of fault classes is done by using machine learning. And Azure ml make this system a self-learning system. Before a failure type detection and predictive maintenance approach can analyze current data, it is necessary to learn from historic data, which has to be labeled with the expected output which is done by using algorithm decision forest. Confusion matrix for predicted classes is shown below:

Predicted Class

	1	2	3	4	5	6
1	94.5%	0.2%	0.8%	2.3%	1.5%	0.7%
2	1.8%	93.9%				4.3%
3	2.2%		95.5%	1.1%	0.7%	0.4%
4	0.9%		0.3%	98.4%	0.1%	0.3%
5	0.6%		0.4%	0.1%	98.2%	0.6%
6	0.2%	0.4%	0.2%	1.1%		98.1%

Figure 22: Confusion Matrix

References

[1] Dheeraj Bansala,David J,Evansb, Barrie Jones, "A real-time predictive maintenance system for machine systems", International Journal of Machine Tools and Manufacture,Volume 44, Issues 7–8, June 2004,

[2] Dheeraj Bansal,David J. Evans," Application of a real-time predictive maintenance system to a production machine system", international journal of machine tools and manufacture 45(10):1210–1221 august 2005

[3] Tomas Olsson1, Elisabeth, Daniel Gillblad , Peter Funk" Fault Diagnosis of Heavy Duty Machines: Automatic Transmission Clutches,

[4] Anders Holst et al,"A Probabilistic Approach to Aggregating Anomalies for Unsupervised Anomaly Detection with Industrial Applications. "Twenty-Eighth International Florida Artificial Intelligence Research Society Conference, May 2015.

[5] Ning Xionget. Al "Fault Diagnosis via Fusion of Information from a Case Stream: In Case-Based Reasoning Research and Development." 23rd International Conference on Case-Based Reasoning (ICCBR-2015), Frankfurt, Germany, September 2015.

[6] Jeff Barnes, "Azure Machine Learning" (Barnes, 2015)

[7] (Valentine Fontama, 2014) (Valentine Fontama, Predictive Analytics with Microsoft Azure Machine Learning, 2014)

Appendix A

A.1 Programmatically access Table storage

A.1.1 Retrieve all entities in a partition

To query a table for all entities in a partition, use a **TableQuery** object. The following code example specifies a filter for entities where 'Smith' is the partition key. This example prints the fields of each entity in the query results to the console.

```
// Retrieve the storage account from the connection string.
CloudStorageAccount storageAccount = CloudStorageAccount.Parse(
   CloudConfigurationManager.GetSetting("StorageConnectionString"));
// Create the table client.
CloudTableClient tableClient = storageAccount.CreateCloudTableClient();
// Create the CloudTable object that represents the "people" table.
CloudTable table = tableClient.GetTableReference("Vehicle");
// Construct the query operation for all customer entities where PartitionKey="Smith".
TableQuery<VehicleEntity> query = new
TableQuery<VehicleEntity>().Where(TableQuery.GenerateFilterCondition("PartitionKey",
QueryComparisons.Equal, "Smith"));
// Print the fields for each customer.
foreach (VehicleEntity entity in table.ExecuteQuery(query))
{
   Console.WriteLine("{0}, {1}\t{2}\t{3}", entity.PartitionKey, entity.RowKey,
      entity.vibration, entity.RPM, entity.Dir );}
```

Appendix B

Screenshots