Predictive Maintenance of Industrial Motors

INTRODUCTION:

Abstract: Industry enables technological trends like Big Data Analytics and Machine Learning techniques to converge into and merge with traditional manufacturing processes, resulting in smart manufacturing. Smart manufacturing techniques leverage the use of Industrial Internet of things (IIoT) technology using IoT sensors that are fitted on physical assets to enhance manufacturing processes. IoT Sensors enable smart manufacturing facilities capable of autonomously exchanging information, which can be used to drive business decisions more accurately. This study proposes an architecture for IIoT based predictive maintenance. A case study from ancillary automobile industry is presented to demonstrate a predictive model for predicting sudden breakdown in industrial machines, thereby enabling the production and maintenance cycle to be 'smart.

INTRODUCTION:

Overview:

Industrial environmental conditions have been upgrading day by day with newly developing automation technology. And, as a result of getting rid of the conventional procedures of manufacturing, this leads to an increase in huge workloads. The next-gen industries will be more advanced and automated as compared with existing ones. This brings a new terminology; "Smart Industries". In this new era, Monitoring as well as controlling of various Industrial applications is challenging as ever. The Internet of Things (IoT), as an emerging technology that brought about rapid advancements in modern technologies, has attracted a lot of attention and is expected to bring benefits to numerous applications. The newly introduced concept is providing a helping hand to achieve Industrial automation through remote access.The aim of the project is to monitor any manufacturing plant remotely for temperature,humidity, vibration, current etc. The program will set the parameters and if the results are not within the parameters, it will send a warning so that immediate action can be taken accordingly.

Purpose:

Predictive maintenance is a technique that uses data analysis tools and techniques to detect anomalies in your operation and possible defects in equipment and processes so you can fix them before they result in failure.It helps in averting unplanned and unnecessary downtime that can affect the company.Predictive maintenance is important in terms of safety of motors and its parts, which include motor controls, motor bearings, etc. For the above reasons we are

LITERATURE SURVEY:

Existing problem:

With the growing adoption of electronics in today's industrial systems, increasing reliability is often hampered by the failings of mechanical components. Reliability is a major issue in today's highly competitive marketplace, the costs associated with unexpected machine failure having potentially drastic effects on a company. Predictive maintenance systems and services are very often unaffordable for SMEs. Therefore there is a need for a low cost means of non-subjective on-line, pre-vibration condition monitoring system for detecting malfunctions in gearboxes, rotating shafts, bearings and similar systems. Currently, only the largest, most critical motors are monitored. Leveraging new, inexpensive RF components and integration techniques, along with advances in microprocessor technology, it is possible to provide an economic solution for wirelessly monitoring motor operating parameters - such as temperature, vibration, current, etc. - for all classes of motors, thus creating enormous potential for energy and cost savings. The proposed LOPTIM technology will surpass the state of the art by utilising the novel combination of stator current analysis, along with the current flow from the inverter to the motor and machinery for continuous pre-vibration monitoring.

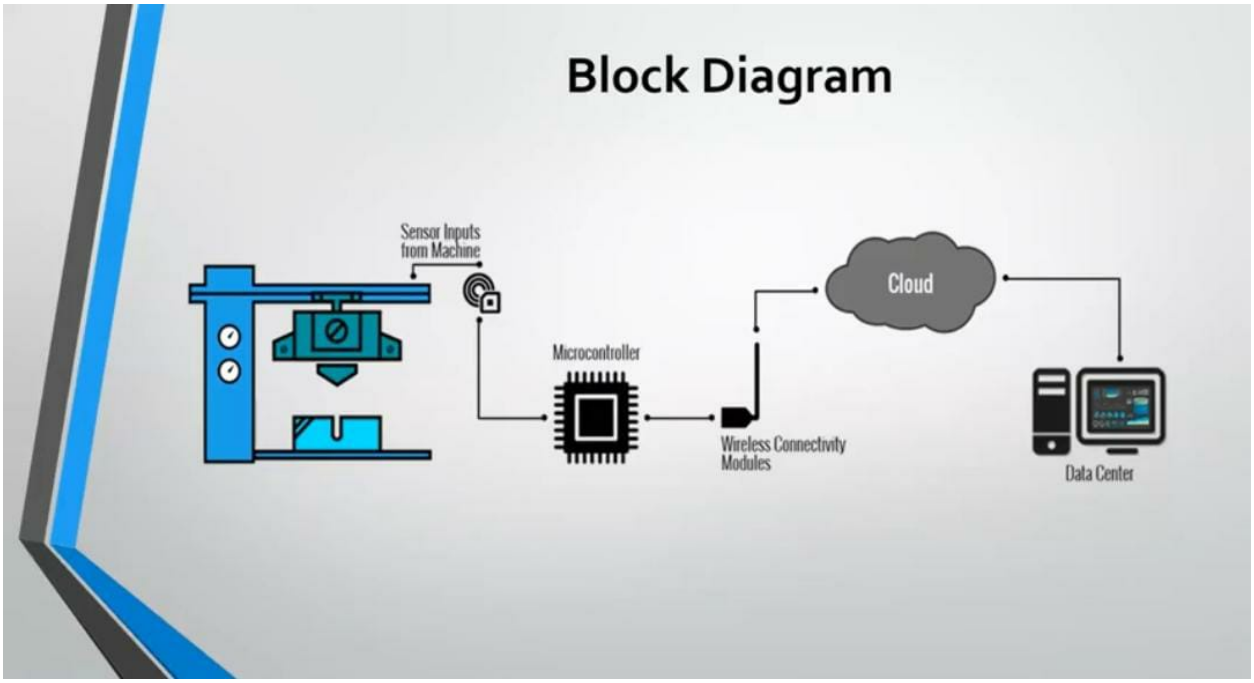
Proposed solution:

Industrial motor predictive maintenance plays a key role in its operation. For proper working conditions, the equipment is continuously observed for its temperature, current, and voltage.In this paper a machine learning model is developed using IBM Auto AI service to predict motor failure. The sensor parameters will be given as an input to the machine learning models which predict the motor failure. The predicted output will be

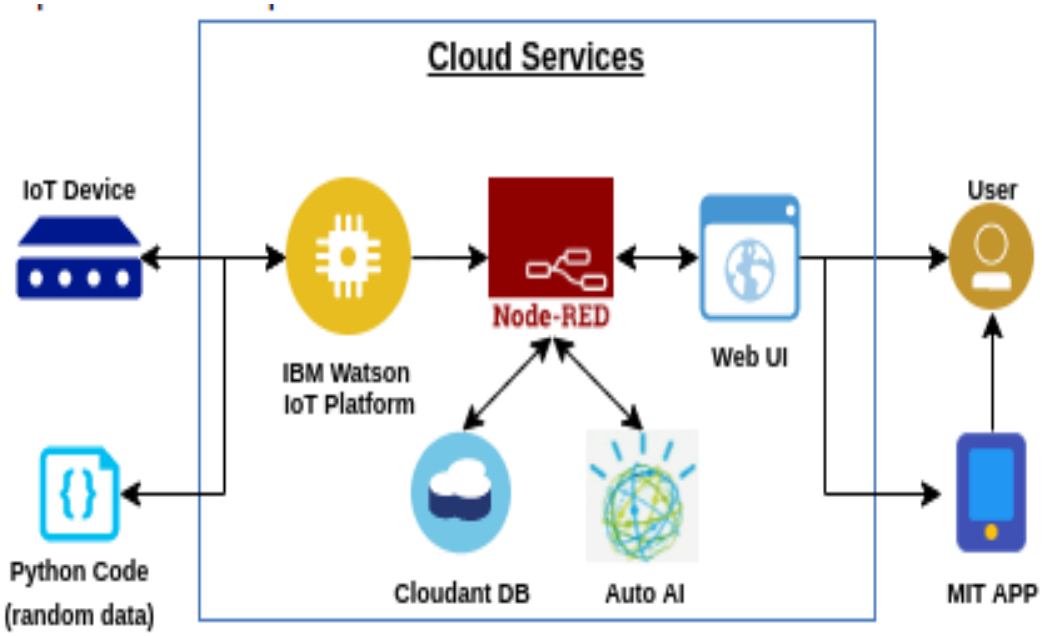
notified to the admins.

THEORITICAL ANALYSIS

Block diagram:



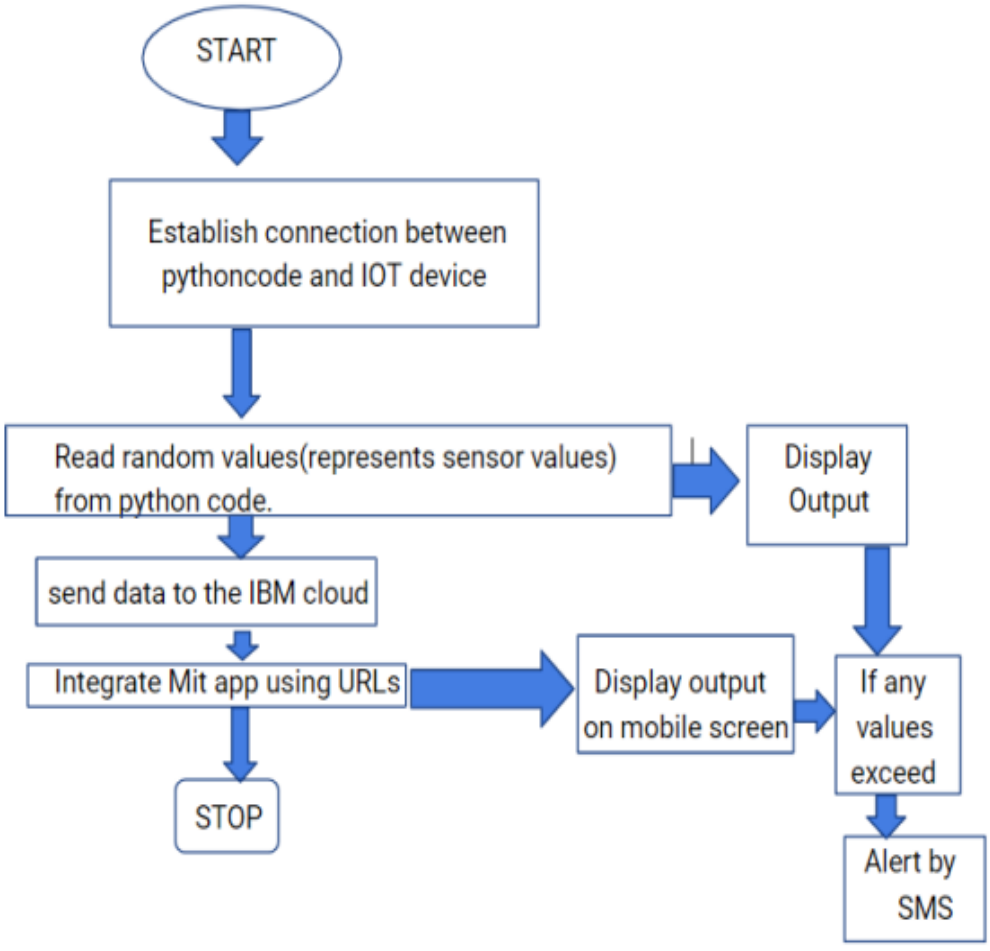
Hardware/Software designing:



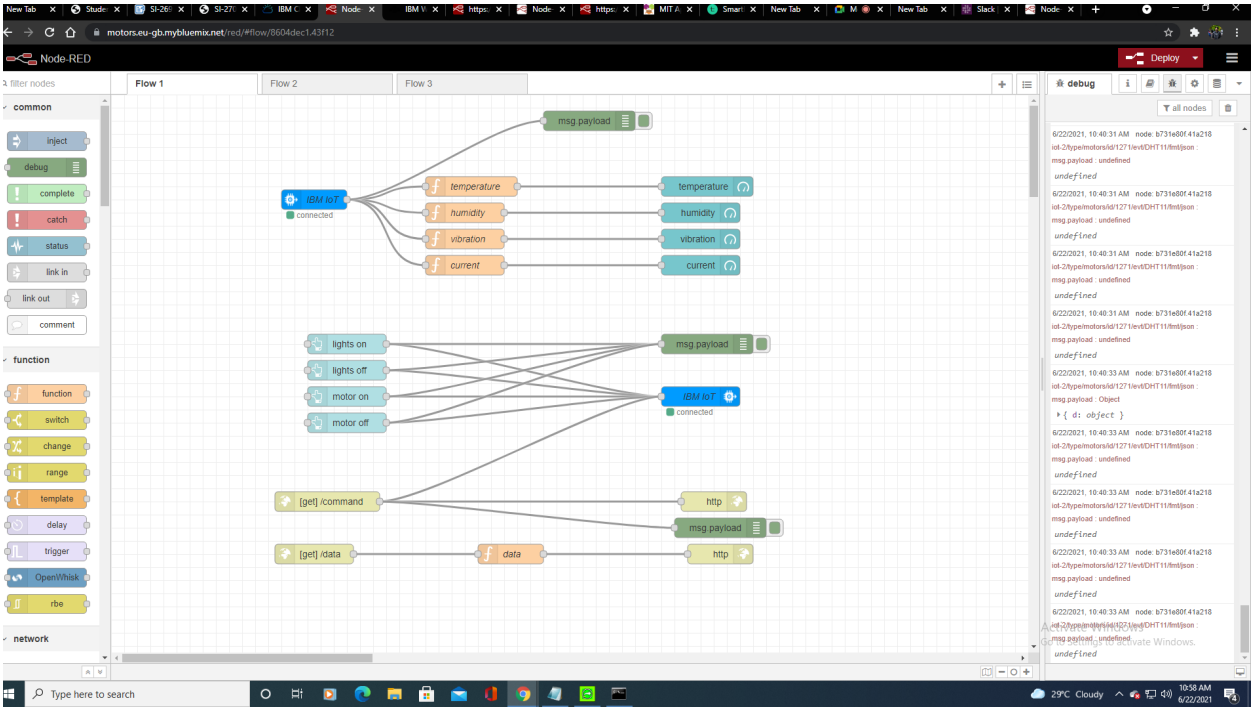
EXPERIMENTAL INVESTIGATIONS

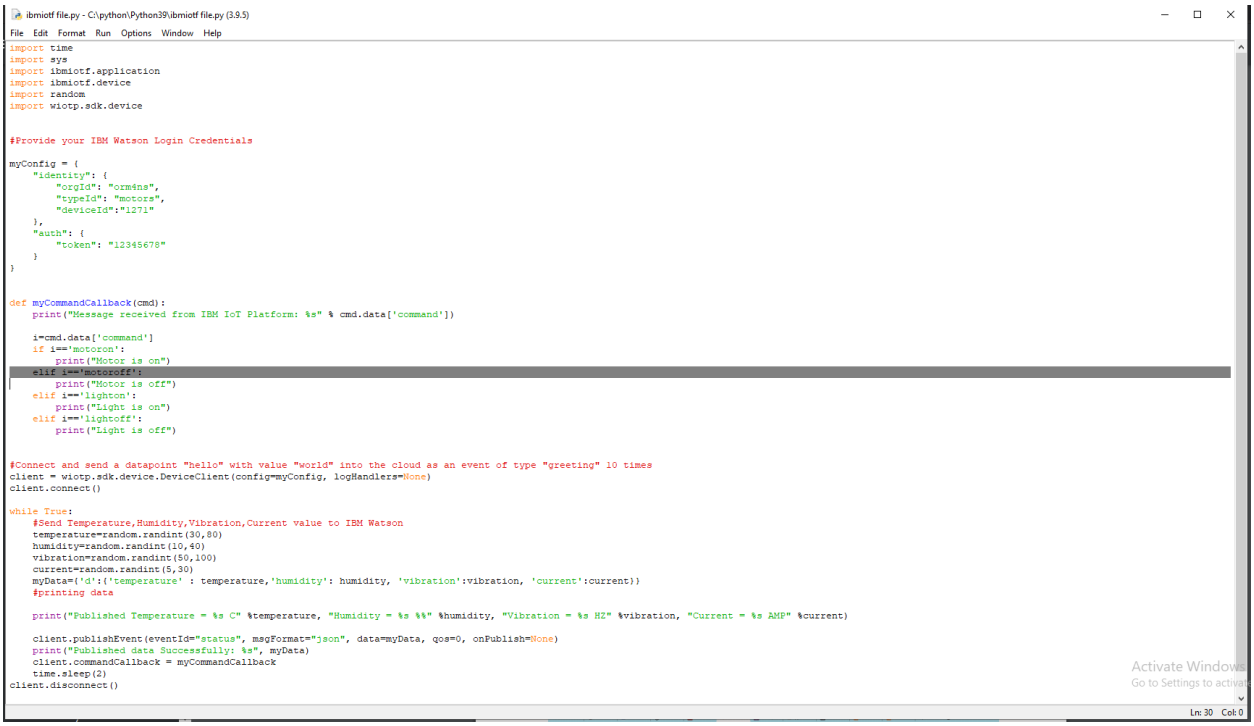
In this project in a Predictive Maintenance of Industrial Motors where the input is taken from the live person. In this paper we are using Python and Internet of Things. In this project we are acessing the random values of employees with their id's, name, skill, time and number of persons entered to the radiation room. As in every few minutes getting alert to the employee that he has to leave from the room, using ibm cloud and node red connecting it to the ibm cloud giving commands to it randomly from the python code and acessing values.The values are processed in the node red by using delay node setting it for fixed seconds delay acessing the id value.

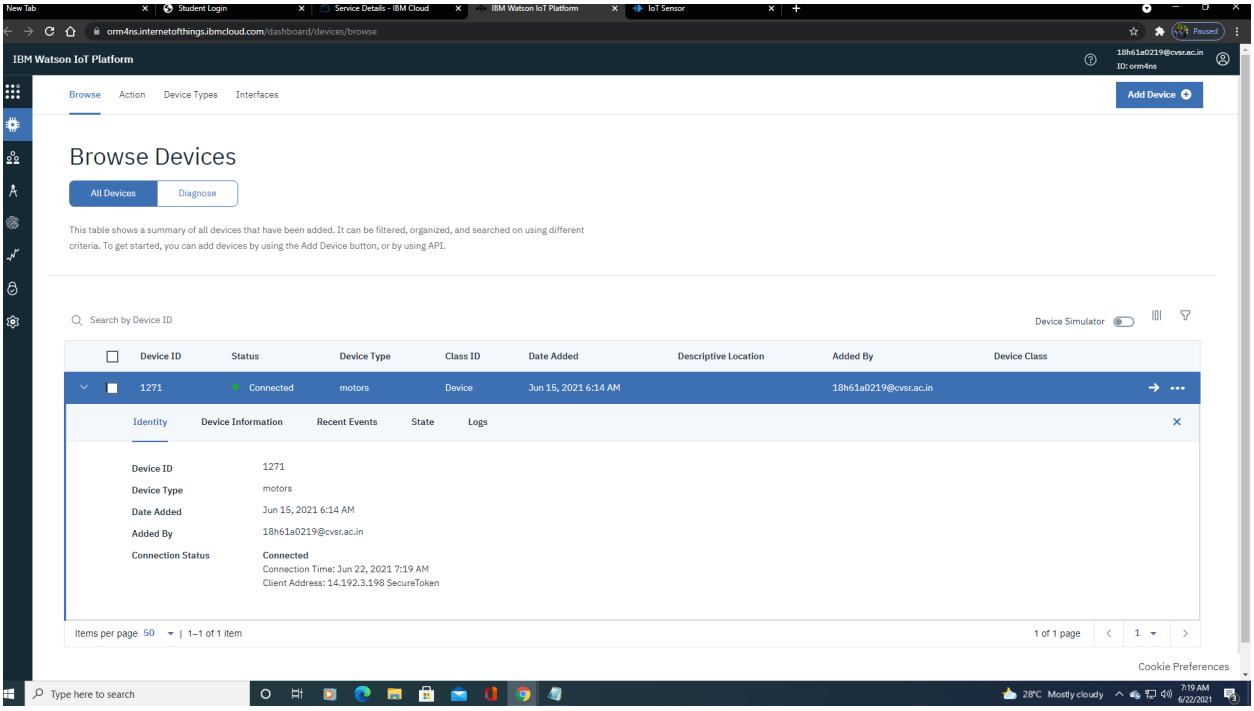
FLOWCHART

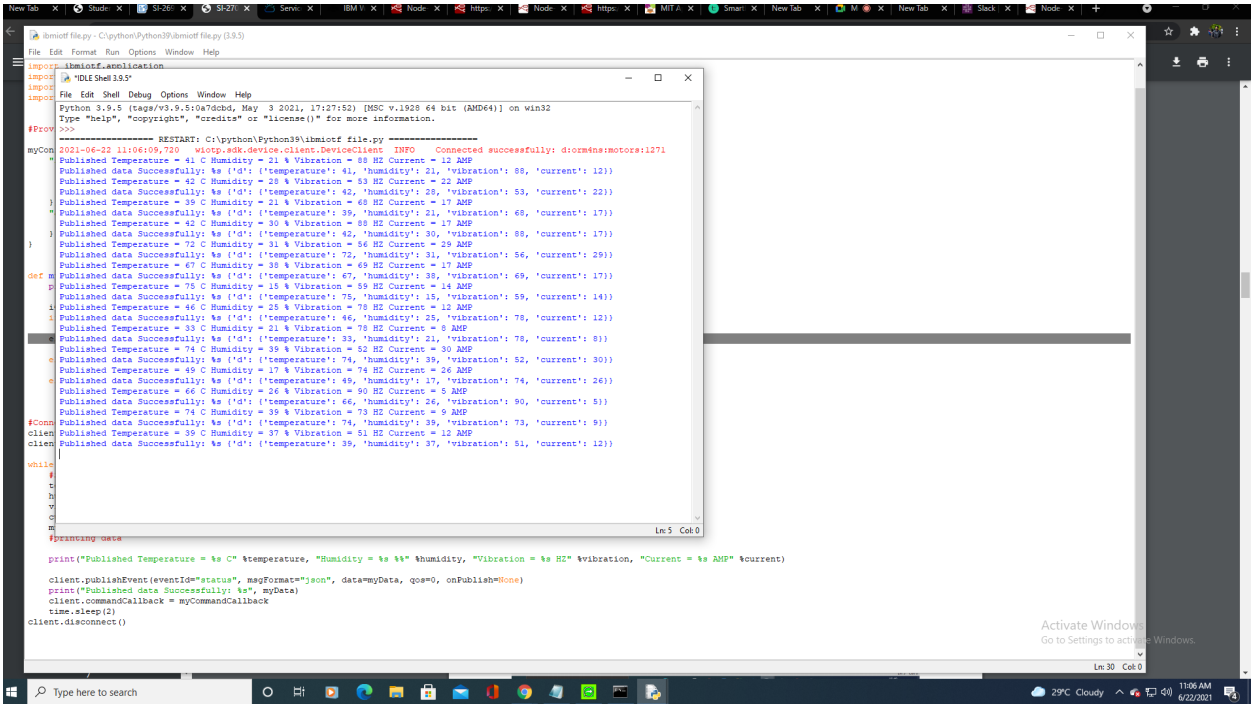


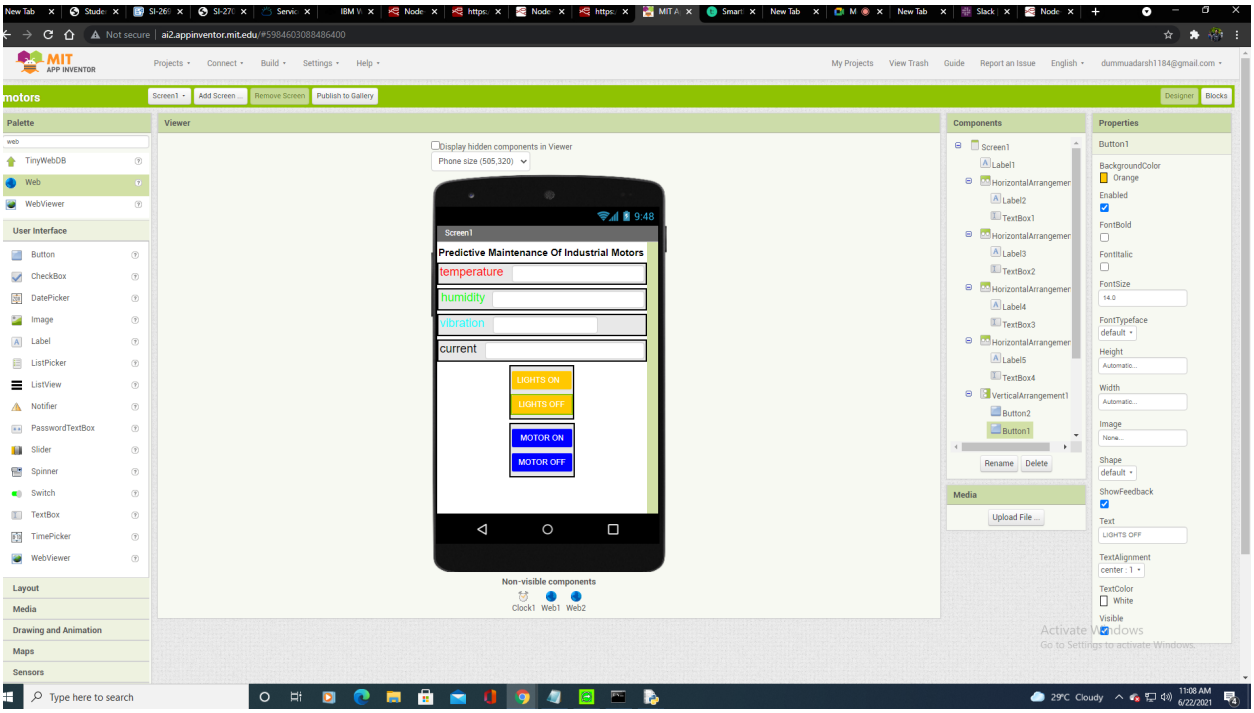
RESULT

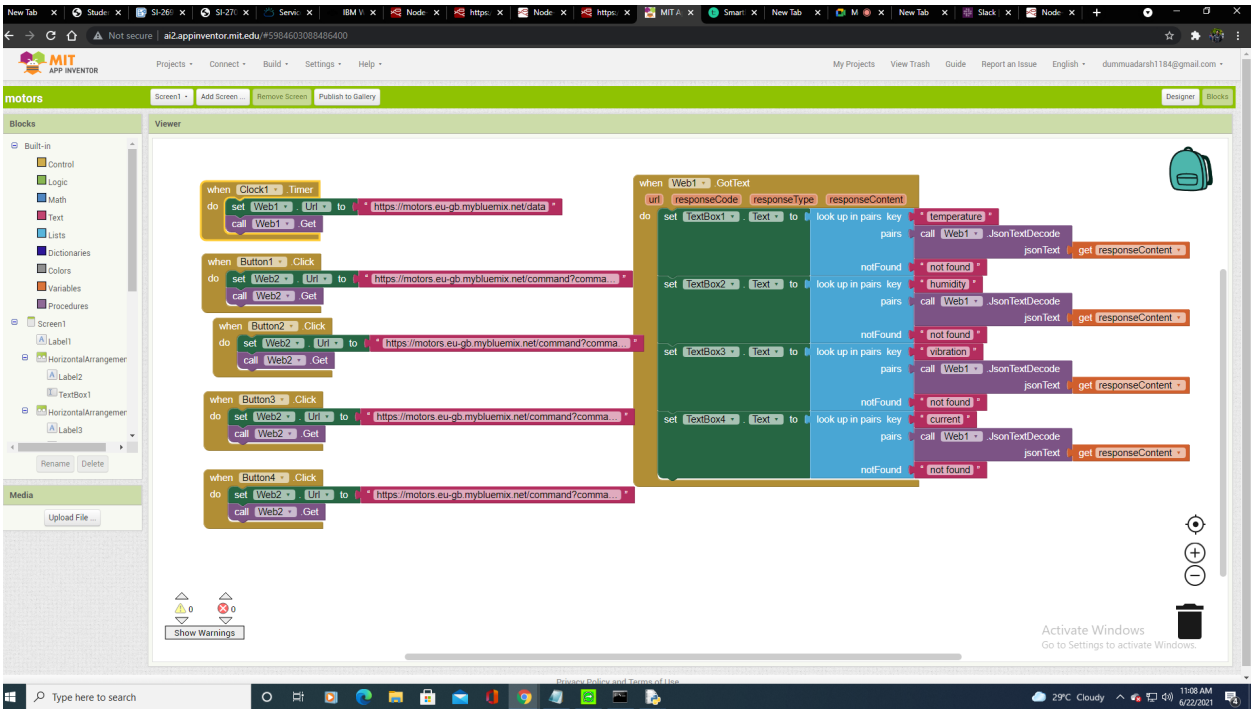

Node-RED
Deploy
Flow 1
Flow 2
Flow 3
debug
common
inject
debug
complete
catch
status
link in
link out
comment
function
function
switch
change
range
template
delay
trigger
OpenWhisk
rbe
network
msg.payload
IBM IoT
connected
temperature
humidity
vibration
current
lights on
lights off
motor on
motor off
[get] /command
[get] /data
data
http
Type here to search
29°C Cloudy
10:38 AM
6/22/2021

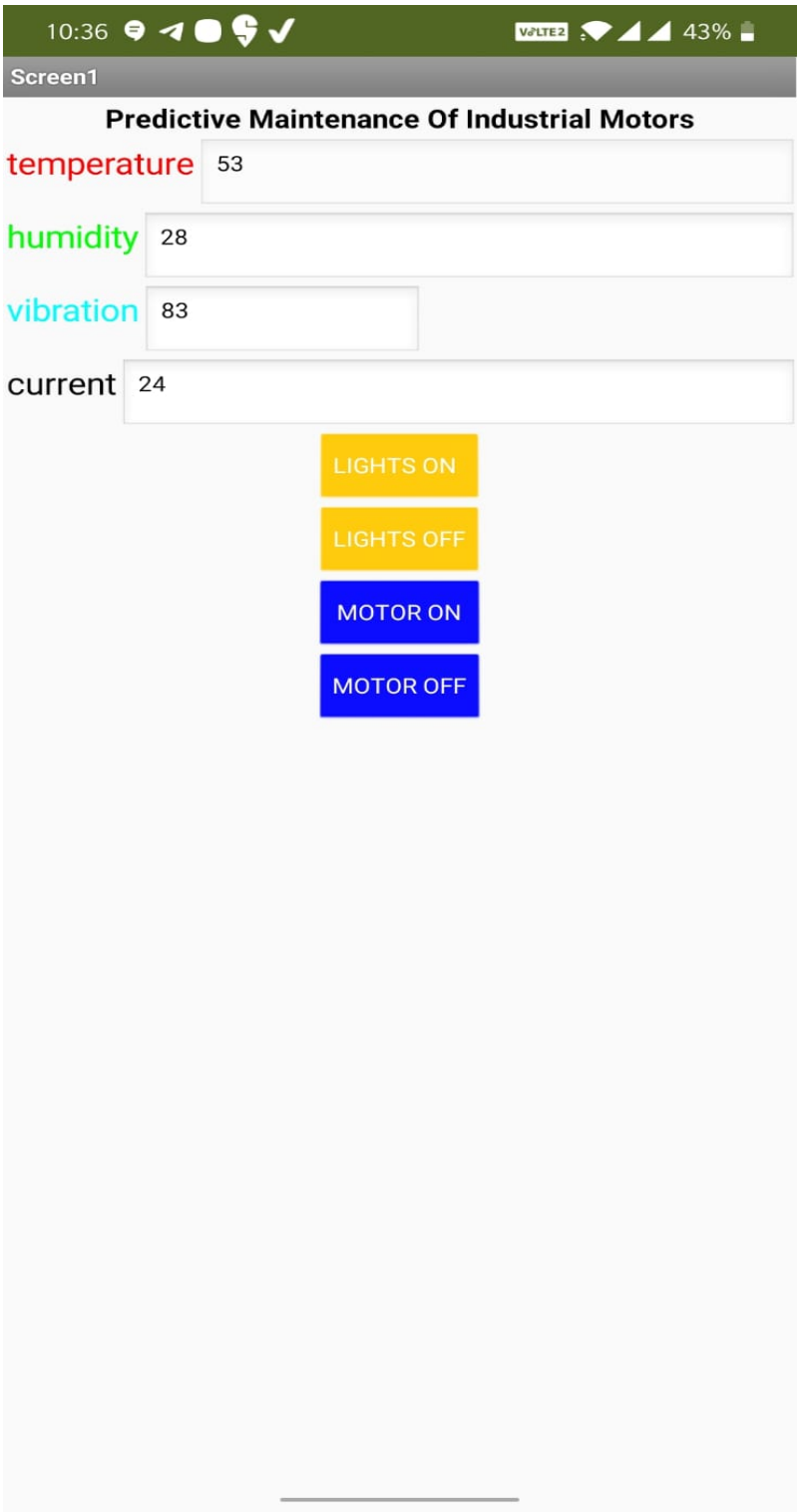

ibmiotf file.py - C:\python\Python39\ibmiotf file.py (3.9.5)
File Edit Format Run Options Window Help
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
import wiotp.sdk.device
#Provide your IBM Watson Login Credentials
myConfig = {
"identity": {
"orgId": "orm4ns",
"typeId": "motors",
"deviceId":"1271"
},
"auth": {
"token": "12345678"
}
}
def myCommandCallback(cmd):
print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
i=cmd.data['command']
if i=='motoron':
print("Motor is on")
elif i=='motoroff':
print("Motor is off")
elif i=='lighton':
print("Light is on")
elif i=='lightoff':
print("Light is off")
#Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()
while True:
#Send Temperature,Humidity,Vibration,Current value to IBM Watson
temperature=random.randint(30,80)
humidity=random.randint(10,40)
vibration=random.randint(50,100)
current=random.randint(5,30)
myData={'d':{'temperature' : temperature,'humidity': humidity, 'vibration':vibration, 'current':current}}
#printing data
print("Published Temperature = %s C" %temperature, "Humidity = %s %%" %humidity, "Vibration = %s HZ" %vibration, "Current = %s AMP" %current)
client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
print("Published data Successfully: %s", myData)
client.commandCallback = myCommandCallback
time.sleep(2)
client.disconnect()
Ln: 30 Col: 0


IBM Watson IoT Platform
Browse
Action
Device Types
Interfaces
Add Device
Browse Devices
All Devices
Diagnose
This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.
Search by Device ID
Device Simulator
Device ID
Status
Device Type
Class ID
Date Added
Descriptive Location
Added By
Device Class
1271
Connected
motors
Device
Jun 15, 2021 6:14 AM
18h61a0219@cvsr.ac.in
Identity
Device Information
Recent Events
State
Logs
Device ID 1271
Device Type motors
Date Added Jun 15, 2021 6:14 AM
Added By 18h61a0219@cvsr.ac.in
Connection Status Connected
Connection Time: Jun 22, 2021 7:19 AM
Client Address: 14.192.3.198 SecureToken
Items per page 50
1–1 of 1 item
1 of 1 page
Cookie Preferences


IDLE Shell 3.9.5
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\python\Python39\ibmiotf file.py ==========
2021-06-22 11:06:09,720 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:orm4ns:motors:1271
Published Temperature = 41 C Humidity = 21 % Vibration = 88 HZ Current = 12 AMP
Published data Successfully: %s {'d': {'temperature': 41, 'humidity': 21, 'vibration': 88, 'current': 12}}
Published Temperature = 42 C Humidity = 28 % Vibration = 53 HZ Current = 22 AMP
Published data Successfully: %s {'d': {'temperature': 42, 'humidity': 28, 'vibration': 53, 'current': 22}}
Published Temperature = 39 C Humidity = 21 % Vibration = 68 HZ Current = 17 AMP
Published data Successfully: %s {'d': {'temperature': 39, 'humidity': 21, 'vibration': 68, 'current': 17}}
Published Temperature = 42 C Humidity = 30 % Vibration = 88 HZ Current = 17 AMP
Published data Successfully: %s {'d': {'temperature': 42, 'humidity': 30, 'vibration': 88, 'current': 17}}
Published Temperature = 72 C Humidity = 31 % Vibration = 56 HZ Current = 29 AMP
Published data Successfully: %s {'d': {'temperature': 72, 'humidity': 31, 'vibration': 56, 'current': 29}}
Published Temperature = 67 C Humidity = 38 % Vibration = 69 HZ Current = 17 AMP
Published data Successfully: %s {'d': {'temperature': 67, 'humidity': 38, 'vibration': 69, 'current': 17}}
Published Temperature = 75 C Humidity = 15 % Vibration = 59 HZ Current = 14 AMP
Published data Successfully: %s {'d': {'temperature': 75, 'humidity': 15, 'vibration': 59, 'current': 14}}
Published Temperature = 46 C Humidity = 25 % Vibration = 78 HZ Current = 12 AMP
Published data Successfully: %s {'d': {'temperature': 46, 'humidity': 25, 'vibration': 78, 'current': 12}}
Published Temperature = 33 C Humidity = 21 % Vibration = 78 HZ Current = 8 AMP
Published data Successfully: %s {'d': {'temperature': 33, 'humidity': 21, 'vibration': 78, 'current': 8}}
Published Temperature = 74 C Humidity = 39 % Vibration = 52 HZ Current = 30 AMP
Published data Successfully: %s {'d': {'temperature': 74, 'humidity': 39, 'vibration': 52, 'current': 30}}
Published Temperature = 49 C Humidity = 17 % Vibration = 74 HZ Current = 26 AMP
Published data Successfully: %s {'d': {'temperature': 49, 'humidity': 17, 'vibration': 74, 'current': 26}}
Published Temperature = 66 C Humidity = 26 % Vibration = 90 HZ Current = 5 AMP
Published data Successfully: %s {'d': {'temperature': 66, 'humidity': 26, 'vibration': 90, 'current': 5}}
Published Temperature = 74 C Humidity = 39 % Vibration = 73 HZ Current = 9 AMP
Published data Successfully: %s {'d': {'temperature': 74, 'humidity': 39, 'vibration': 73, 'current': 9}}
Published Temperature = 39 C Humidity = 37 % Vibration = 51 HZ Current = 12 AMP
Published data Successfully: %s {'d': {'temperature': 39, 'humidity': 37, 'vibration': 51, 'current': 12}}
#printing data
print("Published Temperature = %s C" %temperature, "Humidity = %s %%" %humidity, "Vibration = %s HZ" %vibration, "Current = %s AMP" %current)
client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
print("Published data Successfully: %s", myData)
client.commandCallback = myCommandCallback
time.sleep(2)
client.disconnect()
Activate Windows
Go to Settings to activate Windows.


MIT APP INVENTOR
motors
Palette
Viewer
Components
Properties
Predictive Maintenance Of Industrial Motors
temperature
humidity
vibration
current
LIGHTS ON
LIGHTS OFF
MOTOR ON
MOTOR OFF
Non-visible components
Clock1 Web1 Web2


MIT APP INVENTOR
motors
Blocks
Viewer
when Clock1 .Timer
when Button1 .Click
when Button2 .Click
when Button3 .Click
when Button4 .Click
when Web1 .GotText
https://motors.eu-gb.mybluemix.net/data
https://motors.eu-gb.mybluemix.net/command?comma...
temperature
humidity
vibration
current
not found
Show Warnings



ADVANTAGES & DISADVNTAGES

Advantages:

- Reduction in maintenance costs.

- Reduction in machine failures.
- Reduced downtime for repairs.
- Reduced stock of spare parts.
- Increased service life of parts.
- Increased production.
- Improved operator safety.
- Verification of repairs.

Disadvantages:

- Increases investment in diagnostic equipment
- Increases investment in staff training
- Savings potential is readily seen by management

CONCLUSION

This paper focused on the problem of carrying out predictive maintenance in a industrial motors and presented the results of the preliminary data analysis and feature selection that were performed on a sample of the collected data. The derived data from loT device gives the status of industrial motors about temperature, humidity, Vibration, Current to and from the equipment is continuously monitored to avoid any short circuits and line breakage.so Predictive Maintenance of Industrial Motors plays a major roll in maintaining industrial production. It is possible to have a successful preventive maintenance program. From a cost reduction viewpoint it is essential, but it does entail risk. In order to minimize risk, preventive maintenance has to be carefully planned and carried out by well-trained and motivated workers.

FUTURE SCOPE

If you could see into the future, you would never miss a production target, endure a safety incident, or have a machine go down. Unfortunately, unless we somehow gain the power of clairvoyance, this fantasy will forever be out of our reach. While we may not be able to see into the future, we can predict it. By adopting a predictive-maintenance (PdM) strategy, you can mine your critical-asset data and identify anomalies or deviations from their standard performance. Predictive maintenance doesn't require an extensive overhaul of your infrastructure.Predictive maintenance can be especially useful in industries where the uptime of critical assets drives the bottom line. This includes large, heavy equipment in oil and gas, and mining operations, as well as critical

machines in continuous-manufacturing operations.

BIBLIOGRAPHY

1.Gupta, Stone, and Stein, "Use of Machine HIPOT testing in Electric Utilities." o-7803- 7180-1 IEEE, 2001 (IEEE Dielectrics and Eletrical Insulation Society)
2.R.M. Tallam, T.G. Habetler, R.G. Harley, "Transient Model for Induction Machineswith Stator Winding Turn Faults." IEEE Transactions on Industry Applications, Vol.38, No. 3, May/June 2002
3.https://developer.ibm.com/recipes/tutorials/ui-dashboard-for-iot-device-data-using-node-red
4.https://www.skf.com/binary/21- 285423/Motor-PdM-primer.pdf
5.Smartbridge tutorial lectures
6.IBM cloud blog APPENDIX

Source code:

```
import
wiotp.sdk.device
                    import time
                    import random
                    myConfig = {
                        "identity": {
                            "orgId": "hj5fmy",
                            "typeId": "NodeMCU",
                            "deviceId":"12345"
                        },
                        "auth": {
                            "token": "12345678"
                        }
                    }

                    def myCommandCallback(cmd):
                        print("Message received from IBM IoT Platform: %s" %
                    cmd.data['command'])
                        m=cmd.data['command']
```

```
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    temp=random.randint(-20,125)
    hum=random.randint(0,100)
    myData={'temperature':temp, 'humidity':hum}
    client.publishEvent(eventId="status", msgFormat="json", data=myData,
qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```

UI Output Screenshot: