Internet of Things

DATA ACQUISITION SYSTEM

PROJECT REPORT:

TATA TECHNOLOGIES LTD.

22^{ND} MAY TO 14^{TH} JULY 2017

PREPARED BY:

RAJAT KUMAR SINGH

MANIPAL INSTITUTE OF TECHNOLOGY, MANIPAL

CERTIFICATE

This is to certify that content of Project report entitled "Internet of Things: Architecture and Application" is a work carried out under my guidance and supervision, by **Rajat Kumar Singh** in partial fulfilment of the requirements for the internship in Tata technologies.

Mr. Narendra Saini	Mr. Vijay Kumar
(Project Supervisor)	(Project Guide)

ABSTRACT

Data acquisition is the process of measuring and analyzing various physical and electrical entities like voltage, current, temperature, pressure etc. A Data Acquisition system consists of sensors, interfacing circuitry, an Analog to Digital Converter and Application Software. Such a system has a wide range of applications including Research and Analysis, Control and automation. Our project aims at acquiring data from an accelerometer and a temperature sensor deployed in an industrial environment. This data is sent to a cloud via the Ethernet, using the lightweight, Machine to Machine MQTT protocol, for analysis. A Raspberry Pi 2 is used for data collection and it is interfaced to the sensors using an Arduino Uno microcontroller. The Arduino Uno also converts the Analog data from the sensors to Digital values. Once sent to the cloud, the data is analyzed to check for any abnormalities in the functioning of the said processes in the industrial environment. This system allows us to monitor the health of the machines in an industry without any human intervention.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to all those who provided me with the opportunity to attempt this report. First and foremost, I would like to thank **Mr. Narendra Saini** for providing me with this opportunity to work as a Project Trainee in Tata Technologies Ltd.

It is also imperative that I thank **Mr. Vijay Kumar**, my mentor, who took time off his busy schedule to aid me in learning and guiding me throughout my experience in Tata Technologies Ltd. Furthermore, I would also like to acknowledge the entire **CEIT department** who constantly supported me with their experience working in the industry.

I extend my gratitude to **Manipal Institute of Technology, Manipal** for the continued support. I would also like to thank **Ms. Anu Suryawanshi** and **Mr. Harshad Ghadge** from HR department for all the help they provided throughout my internship.

Once again, I thank everybody who has contributed in making this Project a success and this journey enlightening for me.

TABLE OF CONTENTS

INTRODUCTION

Today, the electronic and computer science field is seeing massive strides in research and innovation. However the mechanical industry is not growing as fast. Hence, in order to improve the efficiency of the engineering processes prevalent in the mechanical industry, companies have forayed into applying the concepts of electronics and computer science in manufacturing industries. Not only does this improve the efficiency of the industries but it also opens up a new field for innovation. Tata Technologies is a pioneer in applying the Internet of Things in the automobile and aerospace sector. The Internet of Things provides as a platform for monitoring the automated processes in industries. This helps in maintaining the health of the machines and controlling any mishaps that are likely to occur. Our project aims at building a system which senses any unusual change in the environment by analyzing real time data and providing feedback about it to the user.

The Internet of Things (IoT) is an important topic in technology industry, policy, and engineering circles and has become headline news in both the specialty press and the popular media. This technology is embodied in a wide spectrum of networked products, systems, and sensors, which take advantage of advancements in computing power, electronics miniaturization, and network interconnections to offer new capabilities not previously possible. An abundance of conferences, reports, and news articles discuss and debate the prospective impact of the "IoT revolution"—from new market opportunities and business models to concerns about security, privacy, and technical interoperability.

The large-scale implementation of IoT devices promises to transform many aspects of the way we live. For consumers, new IoT products like Internet-enabled appliances, home automation components, and energy management devices are moving us toward a vision of the "smart home'', offering more security and energy efficiency.
Other personal IoT devices like wearable fitness and health monitoring devices and network enabled medical devices are transforming the way healthcare services are delivered. This technology promises to be beneficial for people with disabilities and the elderly, enabling improved levels of independence and quality of life at a reasonable cost. IoT systems like networked vehicles, intelligent traffic systems, and sensors embedded in roads and bridges move us closer to the idea of "smart cities'', which help minimize congestion and energy consumption. IoT technology offers the possibility to transform agriculture, industry, and energy production and distribution by increasing the availability of information along the value chain of production using networked sensors. However, IoT raises many issues and challenges that need to be considered and addressed in order for potential benefits to be realized.

A number of companies and research organizations have offered a wide range of projections about the potential impact of IoT on the Internet and the economy during the next five to ten years. Cisco, for example, projects more than 24 billion Internet–connected objects by 2019; Morgan Stanley, however, projects 75 billion networked devices by 2020. Looking out further and raising the stakes higher, Huawei forecasts 100 billion IoT connections by 2025. McKinsey Global Institute suggests that the financial impact of IoT on the global economy may be as much as $3.9 to $11.1 trillion by 2025. While the variability in predictions makes any specific number questionable, collectively they paint a picture of significant growth and influence.

Some observers see the IoT as a revolutionary fully–interconnected "smart" world of progress, efficiency, and opportunity, with the potential for adding billions in value to industry and the global economy. Others warn that the IoT represents a darker world of surveillance, privacy and security violations, and consumer lock–in. Attention-grabbing headlines about the hacking of Internet-connected automobiles, surveillance concerns stemming from voice recognition features in “smart” TVs, and privacy fears stemming from the potential misuse of IoT data have captured public attention. This “promise vs. peril” debate along with an influx of information though popular media and marketing can make the IoT a complex topic to understand.

TATA TECHNOLOGIES

Tata Technologies is a global company in the Tata Group that provides services in engineering and design, product lifecycle management, manufacturing, product development, and IT service management to automotive and aerospace original equipment manufacturers and their suppliers.

The company is a strategic partner for developing complete vehicles, engineering subsystems and components, managing the new product introduction (NPI) process through collaborative engineering tools, such as Product Lifecycle Management (PLM) and tying together information created and used throughout the extended manufacturing enterprise.

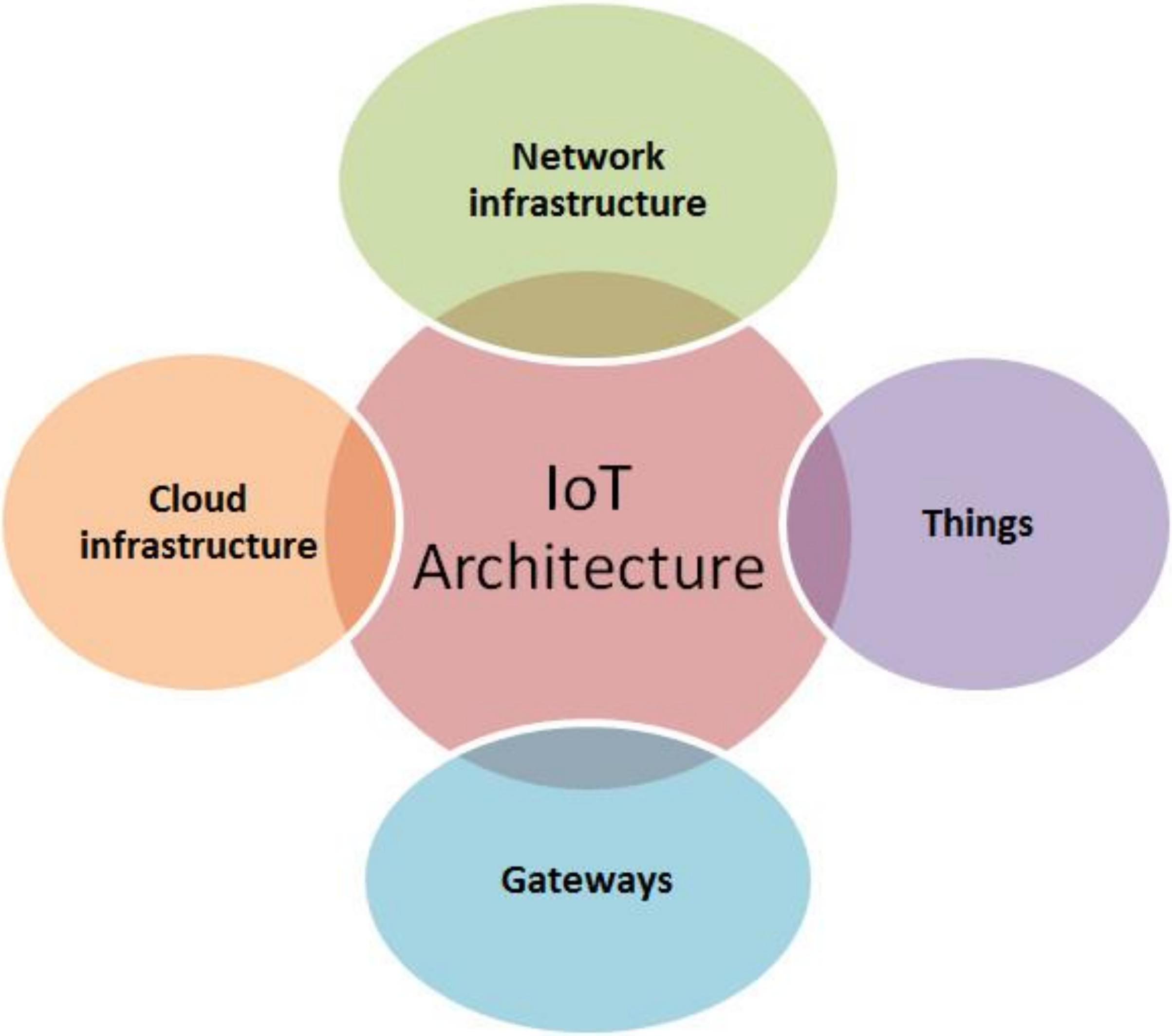
Tata Technologies is headquartered in Singapore, with regional headquarters offices in the United States (Novi, Michigan), India (Pune) and the UK (Coventry) with a combined global work force of more than 8,500 employees serving clients worldwide from facilities in North America, Europe and the Asia-Pacific region.

Purpose
To make product development dreams become reality.

Mission
To transform product development through deep industry knowledge combined with intelligently different approaches to technology, process, innovation and execution

Vision
To bring better products to market for our customers and for the millions of people around the world who benefit from them.

The Internet of Things



WHAT IS THE INTERNET OF THINGS

- The Internet of Things in essence is the connection of physical devices to the internet so that they can interact with each other to achieve a certain goal.
- The interconnection of these embedded devices, is expected to usher in automation in nearly all fields.
- Applications:
    1. Home Automation

2. Bio Sensors and implants which send real time information about the person.
3. In the manufacturing sector, IoT can be used to monitor the functioning of the machines on an assembly line and notify if there are any faults or errors in the operation.
4. Recently for the Champion's Trophy, it was announced that the players will use electronic chips in their bats to monitor their technique. This is also an application of IoT.

    The above are only a few examples, but the Internet of Things has its application in almost every domain thus making it a sought after field of work.

- We use these capabilities to query the state of an object and to change it if possible.
- IoT is a combination of many types of technologies that work together in synchrony.

The term "Internet of Things" (IoT) was first used in 1999 by British technology pioneer Kevin Ashton to describe a system in which objects in the physical world could be connected to the Internet by sensors. Ashton coined the term to illustrate the power of connecting Radio-Frequency Identification (RFID) tags used in corporate supply chains to the Internet in order to count and track goods without the need for human intervention. Today, the Internet of Things has become a popular term for describing scenarios in which Internet connectivity and computing capability extend to a variety of objects, devices, sensors, and everyday items.

While the term "Internet of Things" is relatively new, the concept of combining computers and networks to monitor and control devices has been around for decades. By the late 1970s, for example, systems for remotely monitoring meters on the electrical grid via telephone lines were already in commercial use. In the 1990s, advances in wireless technology allowed "machine–to–machine" (M2M) enterprise and industrial solutions for equipment monitoring and operation to become widespread. Many of these early M2M solutions, however, were based on closed purpose–built networks and proprietary or industry–specific standards, rather than on Internet Protocol (IP)–based networks and Internet standards.

Using IP to connect devices other than computers to the Internet is not a new idea. The first Internet "device"—an IP–enabled toaster that could be turned on and off over the Internet—was featured at an Internet conference in 1990. Over the next several years, other "things" were IP–enabled, including a soda machine at Carnegie Mellon University in the US and a coffee pot in the Trojan Room at the University of Cambridge in the UK (which remained Internet–connected until 2001). From these whimsical beginnings, a robust field of research and development into "smart object networking" helped create the foundation for today's Internet of Things.

OBJECTIVE

Our project aims at developing an Internet of Things based smart system using the concepts of Data Acquisition, Data Transmission and Data Analysis. This involves algorithms to acquire the data in a desired format and processing it to a usable form. Communication plays an important role in this procedure where the data needs to be transmitted in a certain format using a known protocol to achieve smooth functioning of the process. The data is collected in a database on a cloud platform in the form of arrays of floating point numbers. This data is run through a set of machine learning algorithms which analyze the data for any anomaly and notifies the user about any unusual activity.

This kind of a system is useful for an industrial environment which operates heavy machinery and there needs to be constant monitoring of the machines to maintain their health without requiring the need for human monitoring. In the age of automation Internet of Things is allowing us to develop such smart systems for every industrial and household necessities.

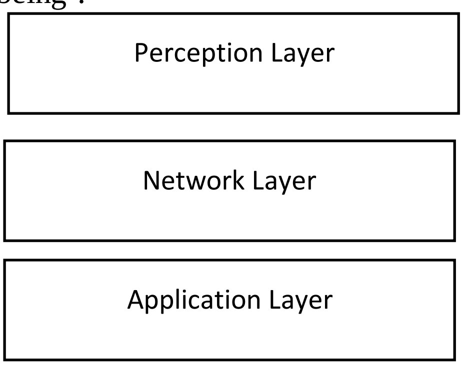
IMPLEMENTATION

We started our project by reading the existing literature on the various architectures and applications of an Internet of Things based system and studying the methodology by which these systems can be made practicable an implemented on a large scale. Following this we studied the variations and different anomalies that can arise in an industrial environment and how they can be identified. There are various factors which govern the functioning of machines in an industry. We observed that there are few physical entities which are absolutely essential in maintaining a healthy industrial environment.
Accordingly we shortlisted a few measurable parameters which are necessary in monitoring the health of an industrial system. Temperature is the most important physical entity in any manufacturing industry and can provide huge amounts of information about the working of the machines. Any sudden change in temperature in an industry is the first indication of a problem which may arise. Secondly the vibration and physical positioning of the machines is imperative and needs to be closely monitored, especially in moving or rotating machines. Hence, an accelerometer which can check the movement of a machine in all directions was used to tell for any faults or fractures in the machinery while being used.
After identifying these essential measurable parameters we observed various faults and errors that could occur in such an environment and tried to simulate the same for parameters like temperature and humidity.
This simulated data was sampled at defined frequency and transmitted over to the cloud platform for further analysis. This allowed us to understand the nuances of the system.
Following this we used sensors and other electronic apparatus to create a prototype of the proposed system and developed code to obtain data from those sensors and analyze the variations observed in this data. An accelerometer and a temperature sensor were used for this purpose.
The proposed system has a wide area of application from the industry to the households and thus forms an integral part of today’s technology.

SIMULATION MODEL

- The data will be read by the sensors at regular intervals of time.
- The sampling frequency of for the sensors will be set by the programmer.
- The data sampled by the sensors will be in the analog form (in terms of voltages or currents) and will be converted to digital using an Analog to Digital converter like ADC 8088.
- In order to model the above data that will be generated, a code can be written to fit the nature of the data approximately by a mathematical function, and this function can be sampled in order to generate discrete values of data.
- In order to achieve a more practicable system noise will be incorporated in the simulation.
- The noise may be due to shutdown, overheating or any other plausible error.
- Further the noise may be added or multiplied to the system to increase its applicability.
- This data will be processed by the CPU and sent over to the cloud to store the data and send notifications to the user device.
- The data will be sent to the cloud using the MQTT (Message Queue Telemetry Transport) protocol.
- The data will be analyzed in the cloud and accordingly the device will be notified.

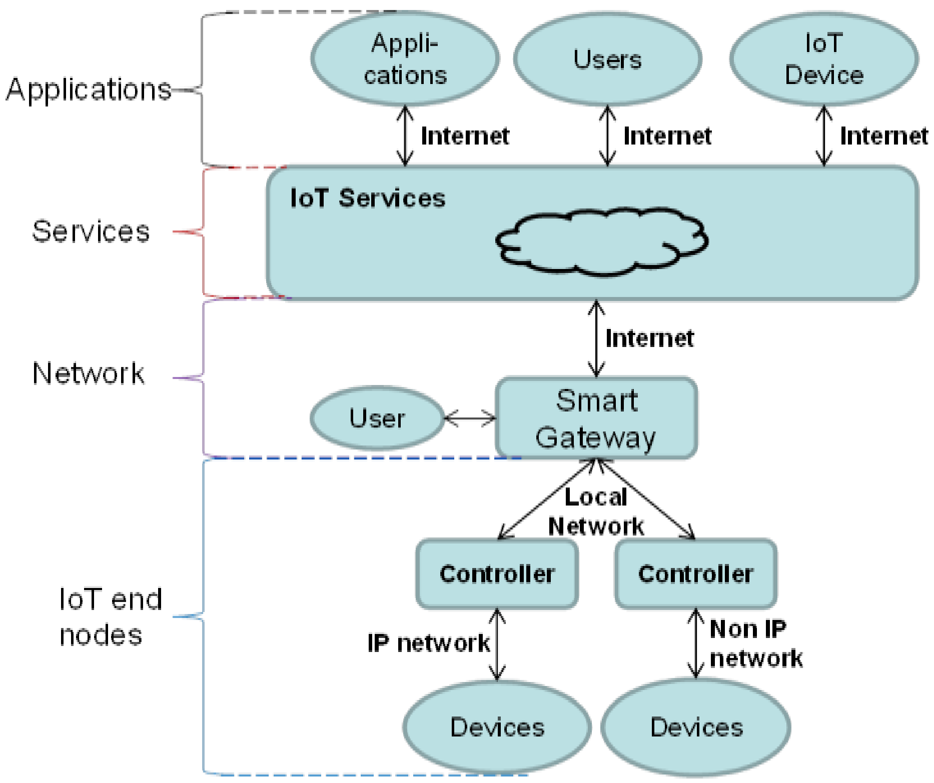
- Sensors: It provides input about its current state (internal state + environment).
- Actuators: It is used to make a change in the environment.
- Architecture: There can be many different architectures for an IoT application. The most basic being[1].



- Perception Layer: This is the device layer of IoT which gives a physical meaning to each object. It consists of data sensors in different forms like RFID tags, IR sensors or other sensor networks [19] which could sense the temperature, humidity, speed and location etc of the objects. This layer gathers the useful information of the objects from the sensor devices linked with them and converts the information into digital signals which is then passed onto the Network Layer for further action.
- Network Layer: The purpose of this layer is receive the useful information in the form of digital signals from the Perception Layer and transmit it to the

processing systems in the Middleware Layer through the transmission mediums like WiFi, Bluetooth, WiMaX, Zigbee, GSM, 3G etc with protocols like IPv4, IPv6, MQTT, DDS etc

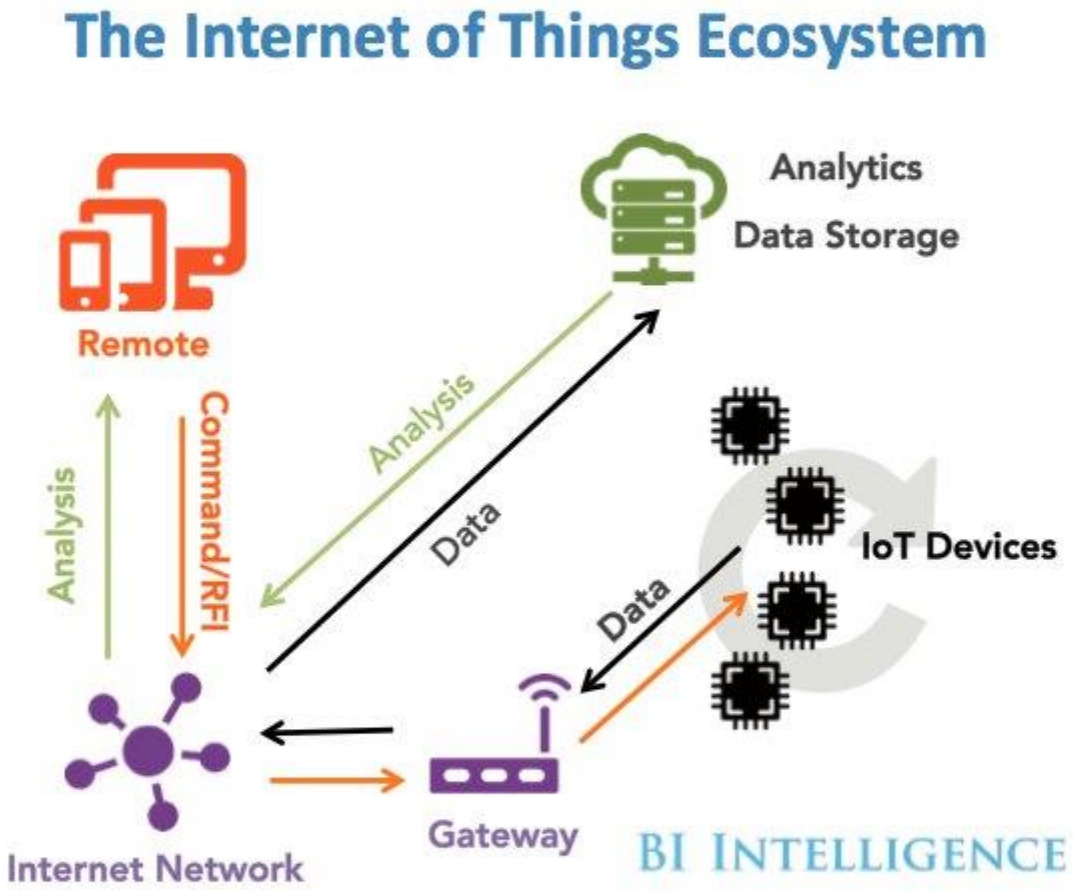
- Application Layer: This layer realizes the applications of IoT for all kinds of industry, based on the processed data. Because applications promote the development of IoT so this layer is very helpful in the large scale development of IoT network. The IoT related applications could be smart homes, smart transportation, smart planet etc.



- In order to simulate the data that will be sent by the sensors we fit a continuous well behaved mathematical function to give the approximate nature of how that parameter behaves.
- The daily temperature can be modelled by a Gaussian curve which has an offset equal to the mean temperature of that region over the day, and the mean and variance will be decided by the maximum and minimum values attained by the temperature over the day.
- For the temperature we can use the Gaussian curve with offset 20, mean 15 and variance 2.1 to f it the temperature range of Pune, as of May 2017.
- All the other sensor parameters can be modelled by mathematical functions in a similar way.
- Noise: In order to incorporate noise in the system, we consider the following reasons for noise to be generated in the system.

1. One of the sensors is faulty and fails to give a reading, thus creating noise in the system.
2. If there is a sudden shutdown of power in the plant, all the readings will be zero.
3. There may also be a continuous noise in the system, it can be additive or multiplicative. This can be due to manufacturing defects, heat generation etc.

- The data will now be sent over to the cloud using the MQTT protocol.
- In the cloud, the data will be processed and analyzed to give appropriate feedback to the user.



DATA ACQUISITION (DAQ) HARDWARE

Transducers sense physical phenomena and provide electrical signals that the DAQ system can measure. For example, thermocouples, RTDs, thermistors, and IC sensors convert temperature into an analog signal that an ADC can measure. Other examples include strain gauges, flow transducers, and pressure transducers, which measure force, rate of flow, and pressure, respectively. In each case, the electrical signals produced are proportional to the physical parameters they are monitoring.

ANALOG INPUTS:

Basic Considerations of Analog Inputs – The analog input specifications can give you information on both the capabilities and the accuracy of the DAQ product. Basic specifications, which are available on most DAQ products, tell you the number of channels, sampling rate, resolution, and input range. The number of analog channel inputs will be specified for both single-ended and differential inputs on boards that have both types of inputs. Single-ended inputs are all referenced to a common ground point. These inputs are typically used when the input signals are high level (greater than 1 V), the leads from the signal source to the analog input hardware are short (less than 15 ft.), and all input signals share a common ground reference. If the signals do not meet these criteria, you should use differential inputs. With differential inputs, each input has its own ground reference. Noise errors are reduced because the common-mode noise picked up by the leads is canceled out.

Sampling Rate: This parameter determines how often conversions can take place. A faster sampling rate acquires more points in a given time and can therefore often form a better representation of the original signal. For example, audio signals converted to electrical signals by a microphone commonly have frequency components up to 20 kHz. To properly digitize this signal for analysis, the Nyquist sampling theorem tells us that we must sample at more than twice the rate of the maximum frequency component we want to detect. So, a board with a sampling rate greater than 40 kS/s is needed to properly acquire this signal.

Multiplexing: A common technique for measuring several signals with a single ADC is multiplexing. The ADC samples one channel, switches to the next channel, samples it, switches to the next channel, and so on. Because the same ADC is sampling many channels instead of one, the effective rate of each individual channel is inversely proportional to the number of channels sampled.

Range: Range refers to the minimum and maximum voltage levels that the ADC can quantize. The multifunction DAQ boards offer selectable ranges so that the board is configurable to handle a variety of different voltage levels. With this flexibility, you can match the signal range to that of the ADC to take best advantage of the resolution available to accurately measure the signal. The range, resolution, and gain available on a DAQ board determine the smallest detectable change in voltage. This change in voltage represents 1 LSB of the digital value, and is often called the code width. The ideal code width is found by dividing the voltage range by the gain times two raised to the order of bits in the resolution. For example, one of our 16-bit multifunction DAQ boards, the AT-MIO-16X, has a selectable range of 0 to 10 or –10 to 10 V and selectable gain of 1, 2, 5, 10, 20, 50, or 100. With a voltage range of 0 to 10 V, and a gain of 100, the ideal code width is 1.5μV Therefore, the theoretical resolution of one bit in the digitized value is 1.5μV.

Critical Considerations of Analog Inputs: Although the basic specifications previously described may show that a
DAQ board has a 16-bit resolution ADC and a 100 kS/s sampling rate, this does not mean you can sample at full speed on all 16 channels and get full 16-bit accuracy. For example, you can purchase products on the market today with 16-bit ADCs and get less than 12 bits of useful data. The most important thing to do is to scrutinize specifications that go beyond the resolution of the DAQ product. National Instruments offers several application notes to help you understand all the specifications on DAQ products. While evaluating DAQ products, also consider the DNL, relative accuracy, settling time of the instrumentation amplifier, and noise, because what you don't know about the DAQ board you are considering can hurt your measurements.
Ideally, as you increase the level of voltage applied to a DAQ board, the digital codes from the ADC should also increase linearly. If you were to plot the voltage versus the output code from an ideal ADC, the plot would be a straight line. Deviations from this ideal straight line are specified as the nonlinearity. DNL is a measure in LSB of the worst-case deviation of code widths from their ideal value of 1 LSB. An ideal DAQ board has a DNL of 0 LSB.
Practically, a good DAQ board will have a DNL of ±0.5 LSB.

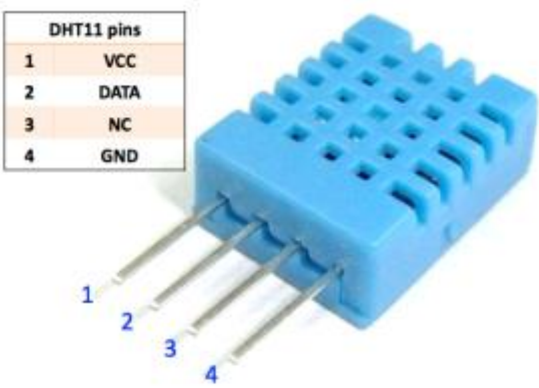
Noise: Any unwanted signal that appears in the digitized signal of the DAQ board is noise. Because the PC is a noisy digital environment, acquiring data on a plug-in board takes a very careful layout on multilayer DAQ boards by skilled analog designers. Simply placing an ADC, instrumentation amplifier, and bus interface circuitry on a one or two-layer board will most likely result in a very noisy DAQ board. Designers can use metal shielding on a DAQ board to help reduce noise. Proper shielding should not only be added around sensitive analog sections on a DAQ board, but must also be built into the layers of the DAQ board with ground planes.

Application Software: An additional way to program DAQ hardware is to use application software. But even if you use application software, it is important to know the answers to the previous questions, because the application software will use driver software to control the DAQ hardware. Application software adds analysis and presentation capabilities to the driver software. Application software also integrates instrument control (GPIB, RS-232, and VXI) with data acquisition. National Instruments offers LabWindows/CVI, application software for the traditional C programmer, and LabVIEW, application software with graphical programming methodology, for developing complete instrumentation, acquisition, and control applications. Both products can be augmented with add-on toolkits for special functionality. Component Works gives complete instrumentation capabilities to Visual Basic through OLE custom controls. For the spreadsheet user, Measure offers DAQ capabilities directly within the Windows Excel environment. For the Windows DAQ user who wants ready-to-run virtual instruments, Virtual Bench offers an oscilloscope, dynamic signal analyzer, function generator, DMM, and data logger. For analysis of the data you collect, we offer HiQ with extensive numerical analysis and visualization capabilities.

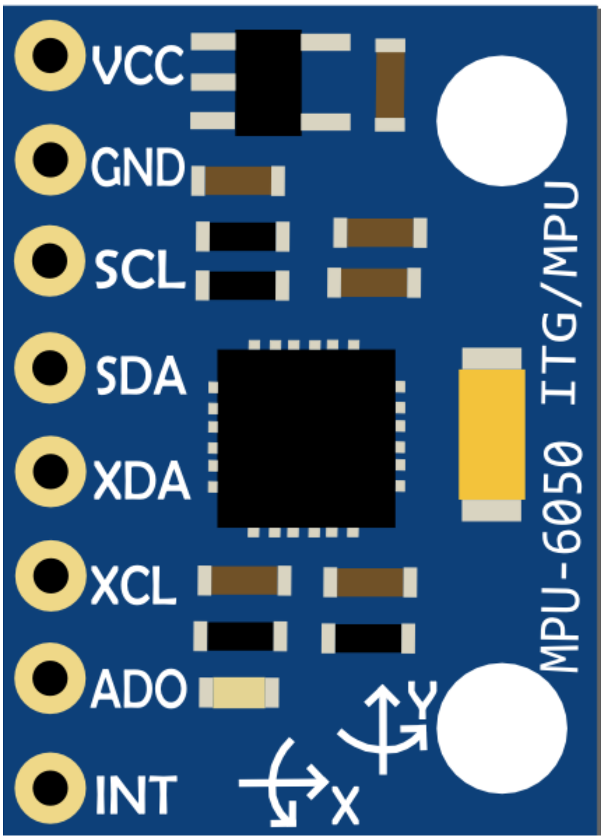
OUR IMPLEMENTATION:

We used the two sensors DHT11 and MPU 6050 as temperature sensor and accelerometer respectively.

The DHT 11 sensor uses a thermistor to convert temperature changes to changes in voltage and has three pins. The pins are for VCC, ground and analog data.



However the accelerometer MPU 6050 has a different way of functioning. It has an accelerometer and a gyroscope measuring six degrees of freedom which are converted to voltages using the DMP, Digital Motion Processor, built in the chip itself. It has two pins to transmit the data, they are SDA and SCL. The SCL pin maintains a clock frequency and the SDA pin carries the data so that this sensor can be interfaced with an asynchronous line of communication. The MPU 6050 uses the I2C (Inter Integrated Circuit) for transmission of data.

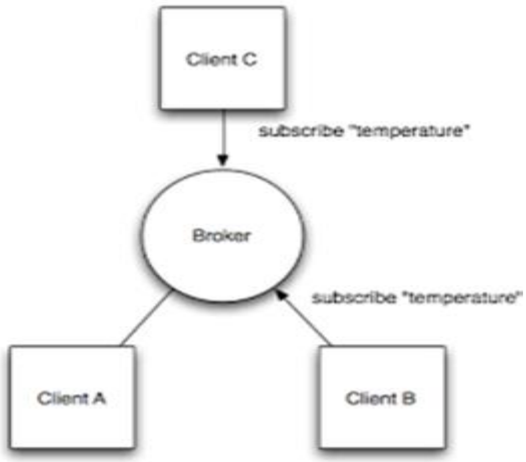


We interfaced these sensors to an Arduino board which was programmed to read the data and give a serial string of data as output for further operation. In order to read the data sent by the accelerometer in the I2C format we used the MPU6050 library by Jeff Rowberg to convert the data into readable values which help in visualizing the movement of the machine and gives the output as the three angles yaw, pitch and roll.

THE MQTT PROTOCOL

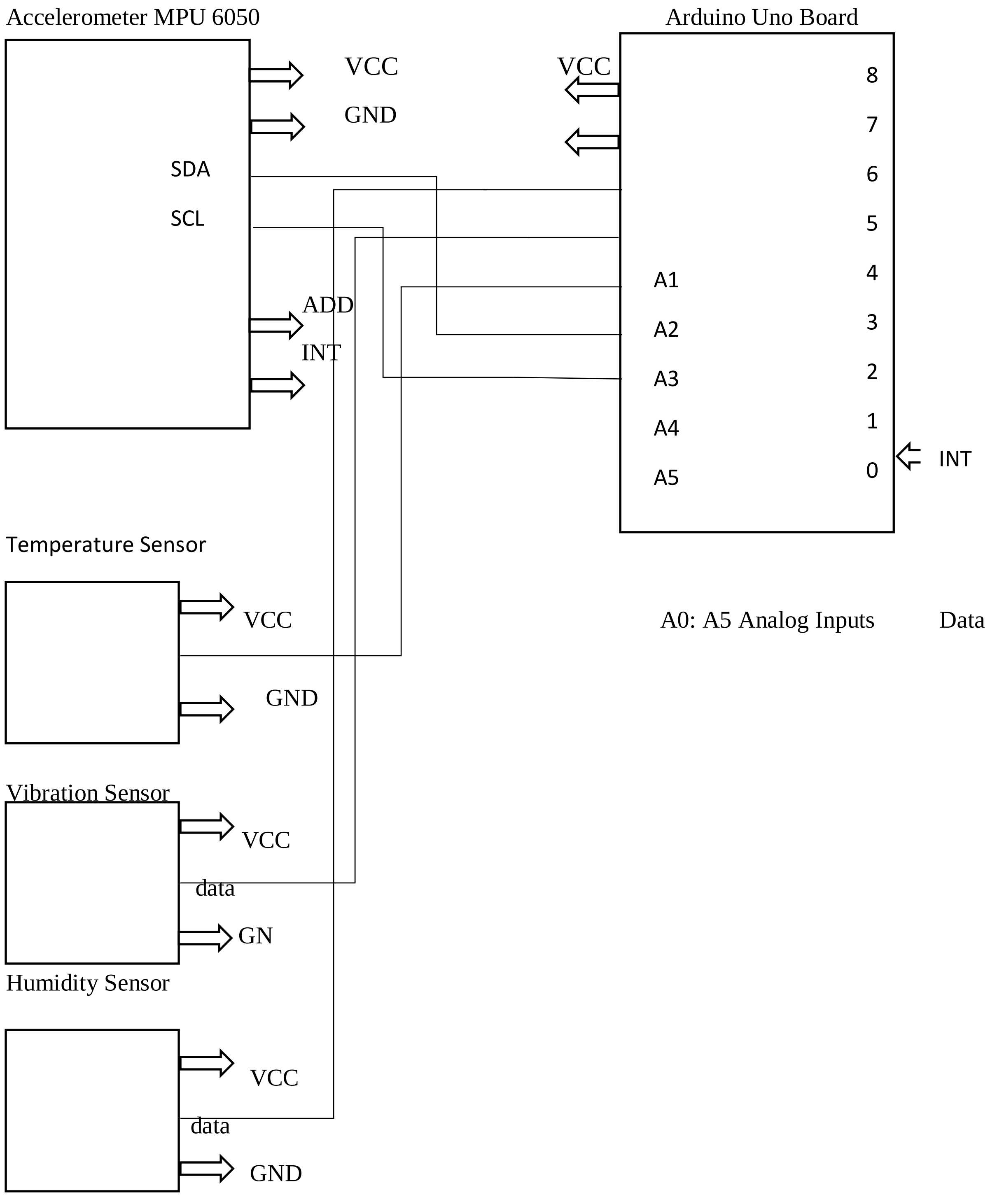
The messaging protocol used in our project is MQTT. MQTT (MQ Telemetry Transport or Message Queue Telemetry Transport) is an ISO standard publish–subscribe based lightweight messaging protocol used on top of the TCP/IP protocol. The main advantage of using this protocol is the minimal network bandwidth requirements. In contrast to the client-server paradigm where the server runs continually to service the client/s , the publish-subscribe approach uses a message broker for distributing messages to interested clients. Instead of initiating connections to specific nodes, information is published and interested parties may subscribe to it. The publish-subscribe paradigm also aims to achieve a scalable multicast, which greatly increases network's efficiency for content delivery when the same content is requested by multiple subscribers. The network also makes sure that subscribers only receive the information they are interested in, effectively preventing most of SPAM and DoS attacks.

Mosquitto is an open source message broker that implements the MQTT protocol and which is used as a broker for our project. The MQTT connection is always between the client and the broker. Clients cannot directly interact with each other. The connection with the broker is handled by the CONNECT/CONNACK messages. Even if the subscribed client has lost its connection with the broker, it will receive the remaining messages once it reestablishes connection with the broker. Scalability and fault tolerance are key aspects of the MQTT protocol.

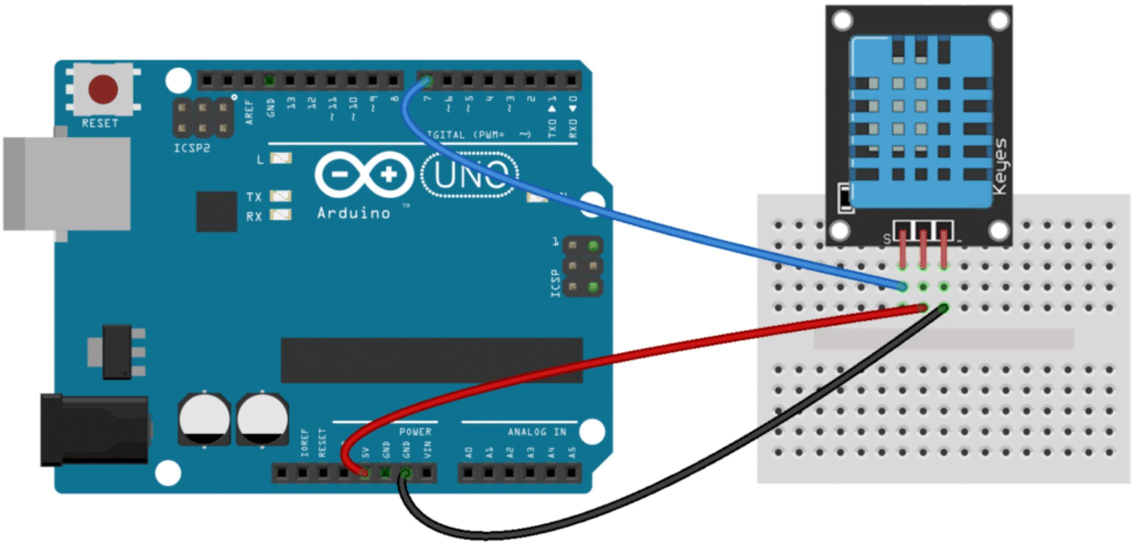


The clients 'A' and 'B' publish their messages on a particular topic name, which in this case is 'temperature'. Client 'C' subscribes to the topic 'temperature' and it receives the messages from both the clients.

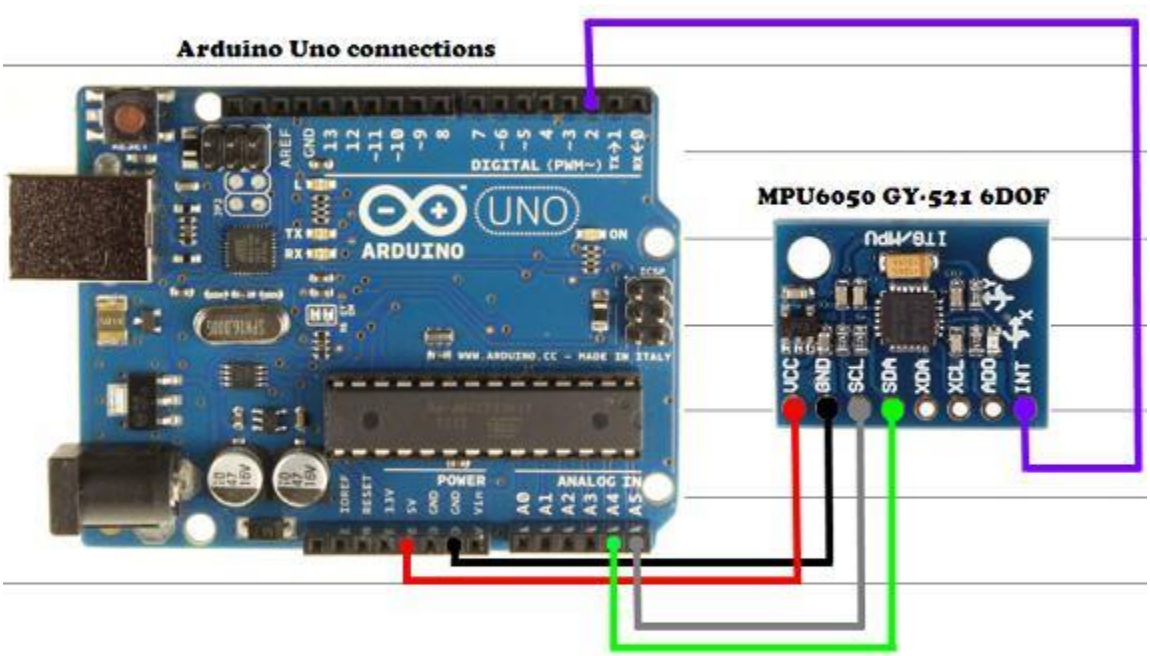
CIRCUIT DIAGRAMS



DHT11 Sensor to Arduino Analog Port



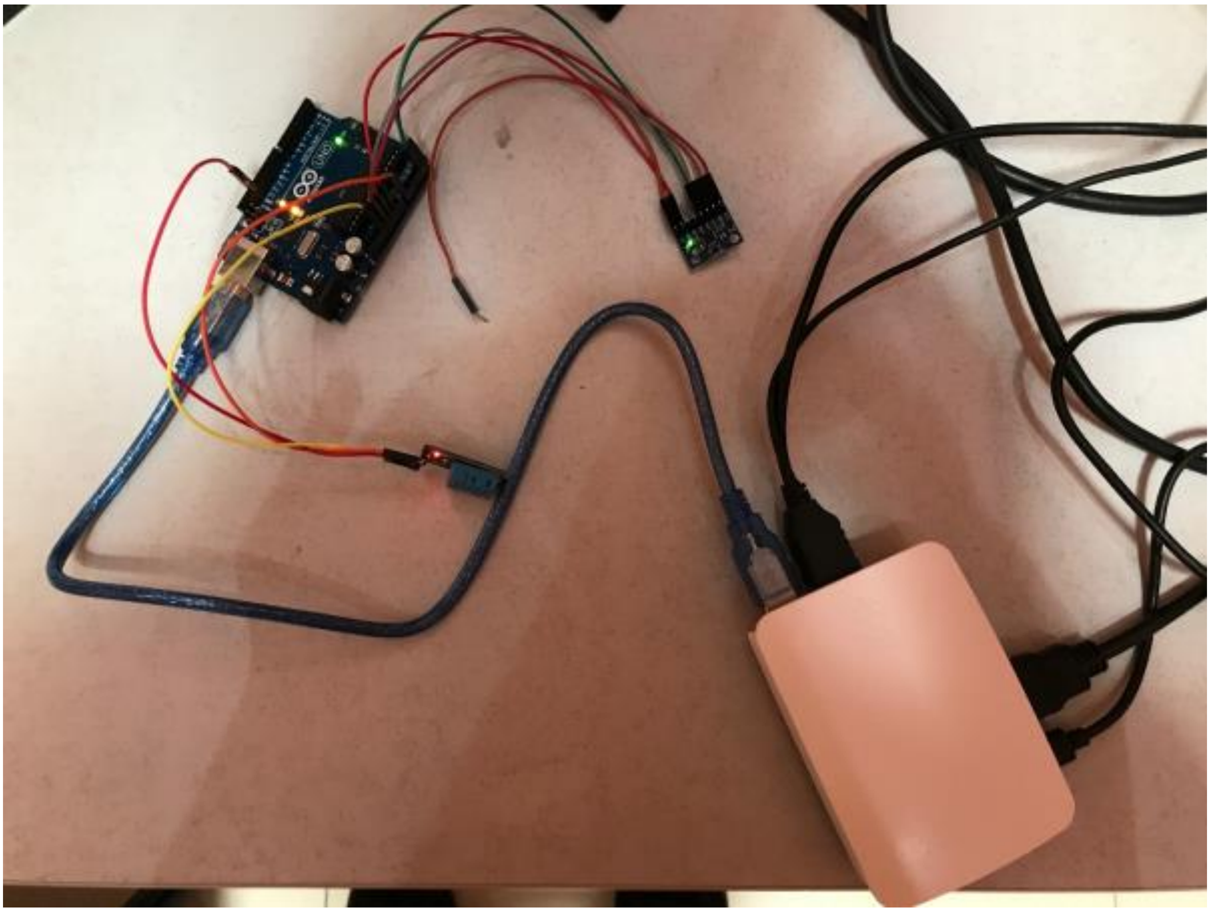
1. MPU6050 Sensor to Arduino Analog Ports (I2C Connection in Port A4 and A5 of Arduino)



The final step of this Data Acquisition process is for the raspberry pi, which acts as the IoT gateway to read the data from the Arduino via a serial USB port and transmit it over to the cloud for storing in the database and further analysis.

The USB serial port of the Raspberry Pi 2 is used to read the data sent from the Arduino. A baudrate of 115200 is used to receive data via the serial port. The sampling frequency at which the data needs to be read is set on the pi. The PySerial library for Python allows us to read the data on the serial port and process it according to our needs. The data is read from the serial port and is packed in the JSON format to be sent to the MQTT broker in real time.

A Python code slows the pi to read the data from the USB port and transmit it over to the cloud using the MQTT protocol. In order to implement the publish-subscribe method of the MQTT protocol we created our own MQTT broker locally which allows data to transmitted to the cloud. Further the data is sent in the JSON format which is a lightweight medium and allows data to be sent in the form of arrays or lists thus making it highly readable.

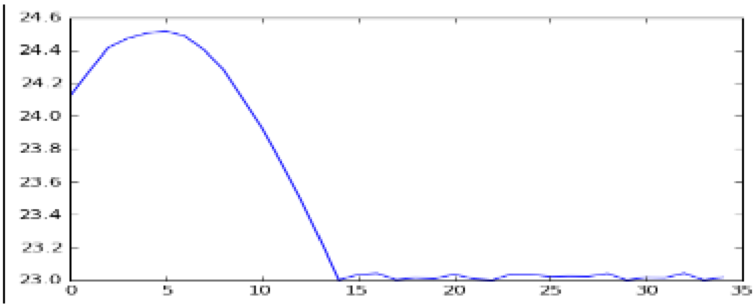


This an image of the real time implementation of our Data Acquisition System prototype. It can be seen how the sensors are interfaced to the Raspberry pi via the Arduino.

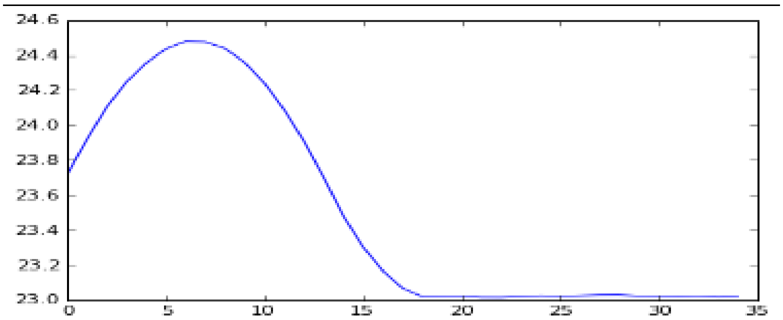
EVENT DETECTION

The sensors transmit their readings via the IOT network to the cloud database. The data used for our experiment is readings from a motion sensor. This motion sensor transmits 3 readings which correspond to the 3 orthogonal axis according to which it is calculated. After preprocessing and normalization, we are left with a set of 3 signals arriving in real time. Our aim is to predict when a particular noise has been introduced in the system, i.e, when the system departs from its normal behavior.

The signals received from the sensor require signal conditioning for better responses when processed by the machine learning algorithm. The sensor readings have minor irregularities or noise which needs to be filtered out. This noise removal is performed effectively by the moving average method over a particular window size. The parameter required for performing this operation is the moving window size. The window size is dependent on the sampling frequency and needs to be altered to prevent loss of information.



INPUT



OUTPUT

The above figure represents the input signal received from the sensor and the output signal received after signal conditioning. This output signal is now fed to the machine learning algorithm.

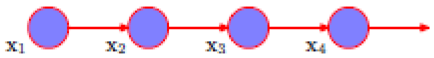
The algorithm used is a supervised learning model. Our primary aim was to identify a particular type of noise. There are two approaches to solve any particular machine learning problem. One is to classify or determine the output based on just the current observations. The second type employs tools to analyze the history of the features and use it to deduce the output. We used the latter as it is more widely used in real-time data and provides a higher accuracy.

The easiest way to treat sequential data would be simply to ignore the sequential aspects and treat the observations independent entities. Such an approach, however, would fail to exploit the sequential patterns in the data, such as correlations between observations that are close in the sequence. In the temporal aspect of a signal, it is possible that a particular state is more likely to be followed by another state. If we treated the signals as independent observations, we ignore this dependency.

The most important task was to convert the set of continuous values of a signal to a discrete form so that it could be used as a feature set. To do this, we extracted the temporal features of the signal. Temporal features would accurately predict the type of noise introduced in a system without depending on the signal level on which the noise is imposed. After transforming out continuous data to a discrete form, out next aim was to map it to a Hidden Markov Model.

Hidden Markov Model:-

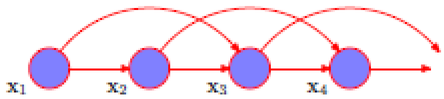
A Markov Model allows for modelling of a state space design in which each space is related to one or more states. A first order Markov Chain is one in which the next observation can be directly predicted by the observation which preceded the current observations.



The joint distribution of this model is given by

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N) = p(\mathbf{x}_1) \prod_{n=2}^{N} p(\mathbf{x}_n|\mathbf{x}_{n-1}).$$

This model is too basic form most application requirements as each state is deterministically associated with another state. A slightly generalized version of this model is the second order Markov chain.



This model is dependent on 2 model and can make a transition to either one of the models.

A further modification and enhancement of this model is the Hidden Markov Model. This model is probabilistic model in which states have a transition probability associated with it and another state. Hence the deterministic nature of the first order Markov chain is eliminated. Each state is known as the observed state as it directly visible. Hidden layer is the output layer corresponding to the output which we want to predict. Each observed state has an emission probability associated with the output layers involved. The transition matrix, emission probabilities and the initial probabilities needs to be specified for predicting the output of a particular state sequence.

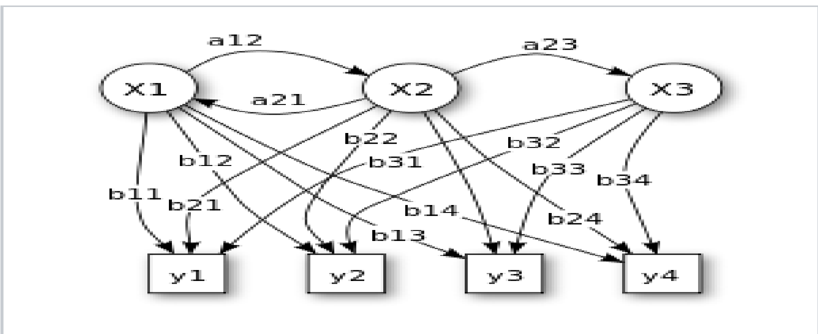


Figure 1. Probabilistic parameters of a hidden Markov model (example)
X — states
y — possible observations
a — state transition probabilities
b — output probabilities

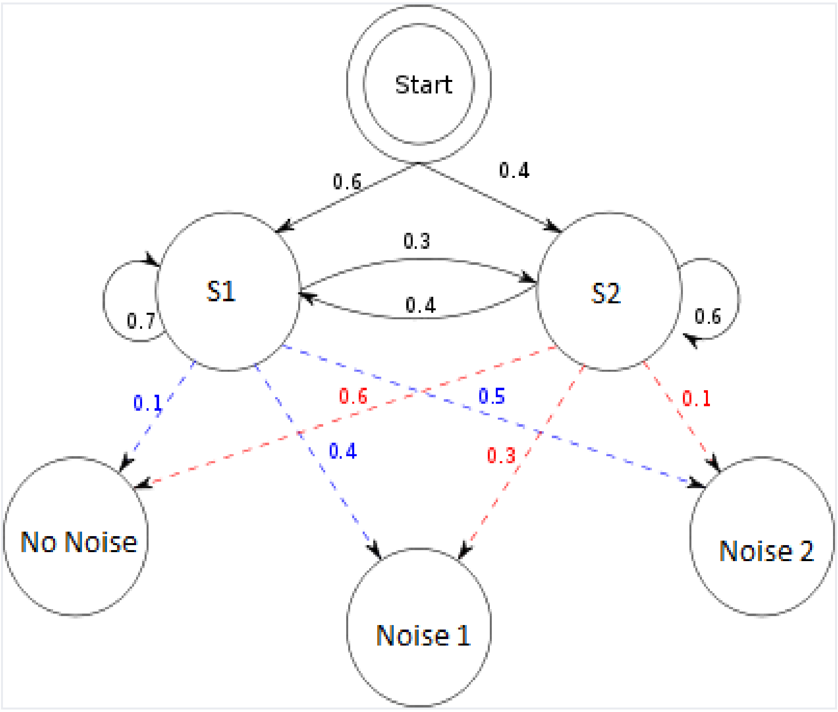
The figure shown above depicts a small model consisting of three observed states with their transitions defined by the variable a. Each observed state has an emission probability for a particular output which is defined by y. In this implementation, the labels of y are the hidden states and are unknown to the user.

Constructing the model:- Extracting features from the continuous data is dependent on the type of data you want to predict. Each output label should have certain parameters associated with itself which distinguishes itself from the rest of the output states.

The first feature we used was the number of slope polarity changes encountered in the recent history. The number of preceding observations to be taken into consideration depends on the persistence of the output state and the sampling frequency in our case. The second feature we used was the number of readings through which data persistently increased or decreased (depends on signal/noise to be detected). The third feature indicated whether there was an increase or decrease in the reading in contrast to the previous input.

The next challenge was to reduce the number of states. For instance, if the recent history consists of 20 sample, the resultant state space would have 20*20*20 (=8000) states in our implementation, which is both unnecessary and redundant as many of the states would hardly occur in a pragmatic setting. In order to reduce the states space, we divided the feature distribution into three ranges after manually analyzing the data. Each of these ranges had an associated label assigned to it. Now we had three possibilities for each of the features which lead to a total of 27 states.

Training the model:- Classical approaches to training the algorithm rely on algorithms such as EM algorithm and Baum Welch algorithm. These algorithms compute the maximum likelihood estimate through which it trains the mode parameters by fitting the model with a state sequence. It takes as an input the number of hidden states which the model consists of. The number usually has to varied for a particular observation sequence to arrive at the best result. For our implementation, we decided to manually compute the transition probabilities and the emission probabilities as the model which we used was supervised and there was no need to guess the hidden states. Each of the hidden states were attributed to a certain noise output to be predicted.

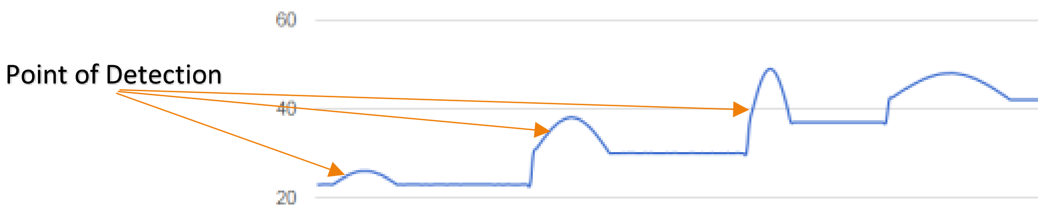


The figure shown above shows a subset of the state space diagram. The hidden states depict whether there is a particular noise. Each state denotes a unique combination of the feature set. The transition matrix is precomputed with the help of supervised learning.

Predicting the output:- In many applications of Hidden Markov models, the latent variables have some meaningful interpretation, and so it is often of interest to find the most probable sequence of hidden states for a given observation sequence. For instance in speech recognition, we might wish to find the most probable phoneme sequence for a given series of acoustic observations. Because the graph for the hidden Markov model is a directed tree, this problem can be solved exactly using the max-sum algorithm. Similarly for our problem we used the Viterbi algorithm to predict the output label of the current state after undergoing a particular sequence of states.

In practice, we are usually interested in finding the most probable *sequence* of states, and this can be solved efficiently using the max-sum algorithm, which in the context of hidden Markov models is known as the *Viterbi* algorithm

Performance of the model:-



In real time if the state sequence of the previous outputs are fed to the prediction algorithm we find that the model optimistically detects the particular noises at the point of detection shown above. All the noise patterns have different amplitude and frequency associated with them. The signal also changes by moving to a different level. There are minor variations in the signal when it is in an isolated state (approximately 0.2). False positives were encountered when these noise patterns demonstrated brief patterns.

To conclude, if the noise patterns to be detected increased then it would accompany a modification in the algorithm, primarily the introduction of another feature. For instance if another noise was introduced which had an increasing slope, it would lead to ambiguity and the affirmation of the noise would have to be done at a later point in the signal, when it can positively be distinguished from the other signal.

CHALLENGES

There are many challenges and considerations in implementing IoT on a massive scale.

1. Data Volume: Traditional analytics tools cannot be applied on the massive amount of data collected. Although data storage is not a major issue because of public clouds , the organization of the data for various analytic purposes like extracting meaningful patterns or detecting aberrant behaviour in the data is of utmost requirement to meet performance standards.

2. Security: As more and more devices become connected, the security of data will become an issue. Data privacy and identity management of devices and individuals is important from the cloud computing point of view.

3. Risks: The development of IoT requires a major need from a capital point of view. In a globally competitive market, companies would not invest in a technology if it is prone to schedule risks, technical risks or performance risks. The mass production of such IoT devices poses such problems which cannot be mitigated because of the presence of unmitigated circumstances.

4. Legacy Systems: Interoperability standardization is a challenge for new IoT devices that need to interface with systems already deployed and operating. IoT engineers are faced with design tradeoffs to maintain compatibility with legacy systems.

5. Data Bandwidth: Large cloud data sets require fast access because of the highly efficient processing elements. The latency for the basic operations need to be decreased for the feasibility of IoT.

CONCLUSION

This project allowed us to learn the various details involved in implementing a smart system on a large scale. It helped us realize the importance of Data Acquisition and Analysis in the real world and the immense scope it has to be applied in various fields.

Our system collects the data and sends it over to the cloud where it is analyzed by running a Hidden Markov Model on the data. This has allowed us to detect any sudden changes in the data which should be uniform in nature and notify the user about the unusual change via an alert that is generated by the system.

We hope this system can be implemented on a large scale and thus can help large industries solve the problem of monitoring and maintaining the structural health of machines.

REFERENCES

1. https://www.ibm.com/blogs/internet-of-things/iot-applications-industries/
2. https://www.ibm.com/developerworks/community/blogs/5things/entry/5_things_to_know_about_mqtt_the_protocol_for_internet_of_things?lang=en
3. www.cse.iitd.ac.in/papers/iot-survey
4. www.cse.wustl.edu/cse570-15/ftp
5. ‘ Getting Started With MQTT A Practical Guide’ , by: Satya Sahoo
6. https://www.internetsociety.org/sites/default/files/ISOC-IoT-Overview-20151221-en.pdf
7. https://www.irjet.net/archives/V3/i12/IRJET-V3I12221.pdf
8. https://www.internetsociety.org/sites/default/files/ISOC-IoT-Overview-20151014_0.pdf
9. http://www.vs.inf.ethz.ch/publ/papers/Internet-of-things.pdf
10. http://onlinepresent.org/proceedings/vol129_2016/41.pdf
11. https://file.scirp.org/pdf/JCC_2015052516013923.pdf
12. http://www.thewhir.com/web-hosting-news/the-four-internet-of-things-connectivity-models-explained
13. http://www.json.org/
14. http://www.internationaljournalssrg.org/IJCSE/2016/Volume3-Issue11/IJCSE-V3I11P108.pdf
15. physweb.bgu.ac.il