**ANOMALY DETECTION IN**

**SMART FACTORIES**

**INTRODUCTION TO IoT**

# COURSE PROJECT REPORT

***Submitted by***

**Name of the students:**

1. **Ummadisetty Dheeraj – 211010253(ECE)**
2. **Govindula Srinith – 211010226(ECE)**



**Dr. Shyama Prasad Mukherjee International Institute of**

**Information Technology, Naya Raipur**

**July-2022**

# Abstract

The likelihood of production failures increases as systems in smart manufacturing get more complicated and generate vast amounts of data. The desire to reduce or eliminate production failures occurs, and anomaly detection is one method for doing so. However, there are several factors to take into account while deploying anomaly detection systems. This paper discusses an overview of the elements, advantages, difficulties, solutions, and unsolved issues of anomaly detection in smart factories

The project aims to solve this problem by creating a device that utilizes an ADXL-345 accelerometer connected to NodeMCU. The device performs fan monitoring continuously. The results are displayed on the ThingSpeak cloud.. With the help of this device, users will be able to detect the Anamolies that present in the Smart factories.

# Introduction

The totally digital revolution is just starting to take hold in the manufacturing sector. An effort is currently being made to integrate cutting-edge technologies like 5G, the Internet of Things (IoT), and cloud computing to take manufacturing to a completely new level—that of smart factories. Finding anomalies is a crucial component of industrial process management. By doing this, one can avoid production interruptions or even damages, which would result in additional financial loss. Future inspection will be totally autonomous, in contrast to existing methods, and reactions will function without the need for human participation by evaluating IoT sensory input.

Anomaly detection is a key idea in the field of data analysis, and it has been extensively studied in a number of areas, including fraud detection, fault detection, and intrusion detection, as well as in a wide range of application domains, including cyber security, healthcare, tax, insurance, finance, traffic management, energy management, automated industrial processes, and many others. Unusual data may reveal important and serious situations that need for immediate attention. A deviation from the norm in the status or behaviour of the IoT system is referred to as an anomaly in smart manufacturing.

* Implementation of IoT based Anamoly detection in smart factories
* The System depends upon the basic use Internet of Things (IoT) approach for above problem.
* To get an estimate amount of data from Normal Ceiling fan and Anamoly fan

# Objectives

.

* Using IOT, this paper shows real time condition monitoring system for Industrial Motors
* To show the difference between the running operation of the healthy and faulty motor

# Proposed Methodology

**3.1 Conceptual diagrams**

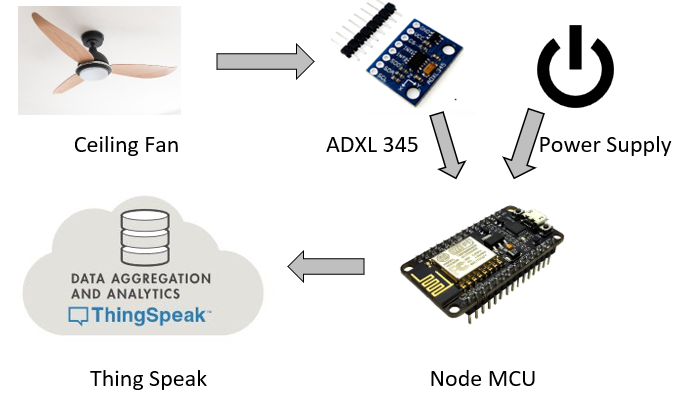


Fig 1 Conceptual Diagram

In the conceptual diagram, Node MCU has been Connected with ADXL 345 Accelerometer.

1. A low-cost System-on-a-Chip (SoC) called the ESP8266 serves as the foundation of the NodeMCU (Node MicroController Unit), an open-source ecosystem for developing both software and hardware. The ESP8266, created and produced by Espressif Systems, includes all of the necessary components of a computer, including a CPU, RAM, networking (WiFi), and even a contemporary operating system and SDK. Because of this, it is a great option for all types of Internet of Things (IoT) projects.
2. The ADXL345 is a tiny 3-axis accelerometer with a dynamic range of +/-16g, 13-bit resolution, a 3200Hz maximum bandwidth, and a 3200 times per second maximum data transfer rate. It is a digital accelerometer sensor that produces three digital acceleration measurements.
3. Now, Node MCU which is having in-built ESP 8266 WI-FI module is connected to the ADXL 345 through jumper wires and these are connected to the fan

The ESP 8266 WI-FI module is interfaced with NodeMCU, connecting to the Internet through a mobile data network. All this data has been sent to the ThingSpeak cloud server, we can have a proper visualization of the data in the form of the graphs

**3.2 Algorithms**

**Step-1**: Select the number K of the neighbors

**Step-2**: Calculate the Euclidean distance of K number of neighbors

**Step-3**: Take the K nearest neighbors as per the calculated Euclidean distance.

**Step-4**: Among these k neighbors, count the number of the data points in each category.

**Step-5**: Assign the new data points to that category for which the number of the neighbor is maximum.

**Step-6**: The data is Classified

**3.3 Working**

* Our IoT device built on Node MCU in this project has a straightforward application.
* Our prototype is placed on the ceiling fan. As soon as the system is powered up, the Node MCU will run the sensor read functions to get the raw data from the ADXL 345 Accelerometer
* We Powered the Node MCU using Powerbank in which both are attached to the Ceiling fan
* We will collect the data of both normal ceiling fan and Faulty fan
* Then the sensor data is sent to ThingSpeak cloud service via inbuilt Esp 8266 WI Fi module and it will connect with the network provider server to connect it to the internet.
* Next, with the help of the ThingSpeak cloud service, we can store the collected data.in a .csv file
* We use this data to predict the anomalies present in the data .

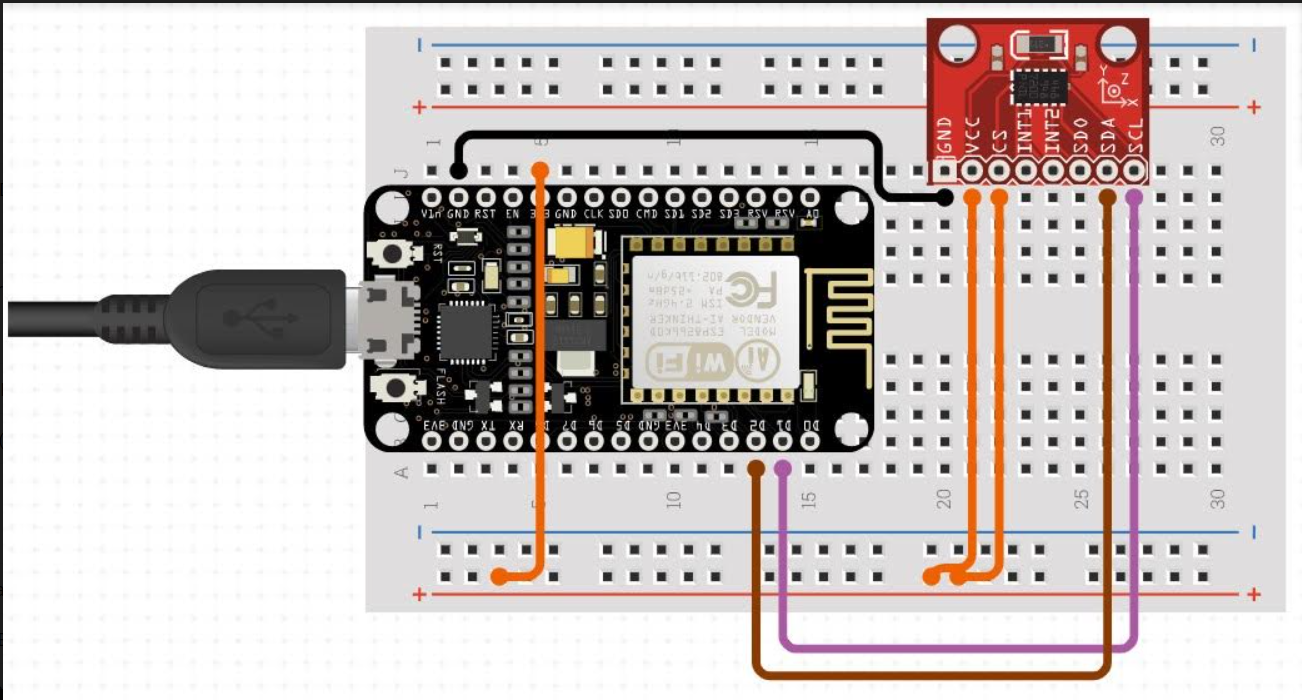


Fig 2 Sensor Node

* 1. **Flowchart**

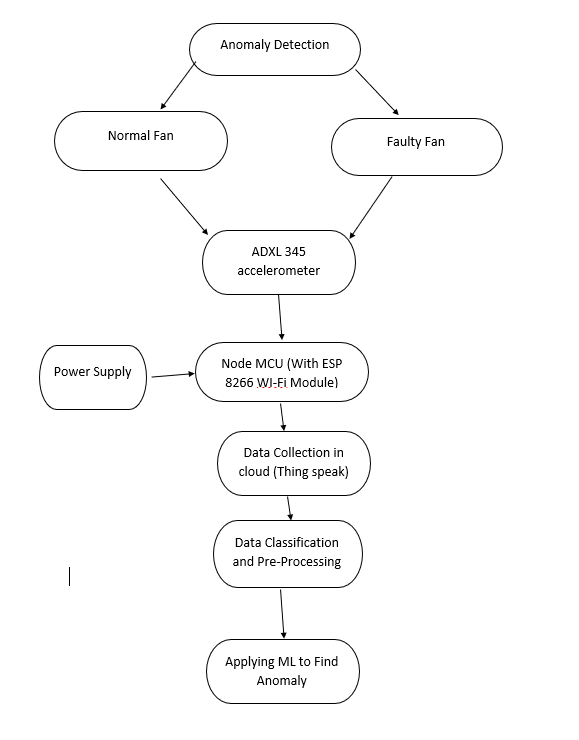


Fig 3 Flowchart

* 1. **Real-Time Implementation**.

\  

Fig 4 Placing on normal fan Fig 5 Placing on Faulty fan

The Fig 4 and Fig 5 are images shows the implementation of prototype on real life in life scenario.

# Results

When the prototype gets completed, the results we will be getting are:

* The sensed data and the calculated data will be stored on the cloud using ThingSpeak in real-time. And we can see this data in and in real-time also.
* Well analyzed vibration signal easily shows the difference between the running operation of the healthy and faulty motor.
* It detects the anomaly points present in the given data.

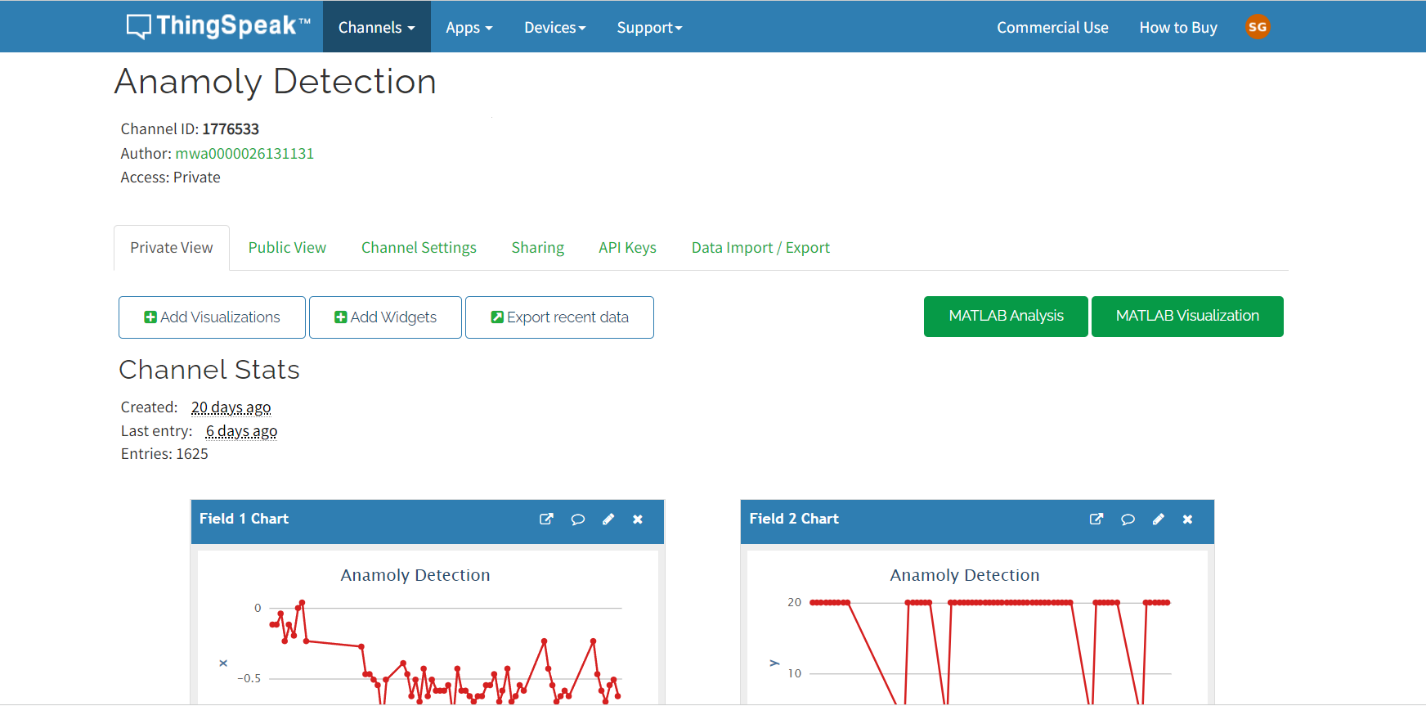


Fig 6. Web view of cloud window

The Figure 7 shows the web view of the ThingSpeak cloud window. There are three fields in this channel.

Field 1,Field2,Field 3 represents the values of acceleration of the data.

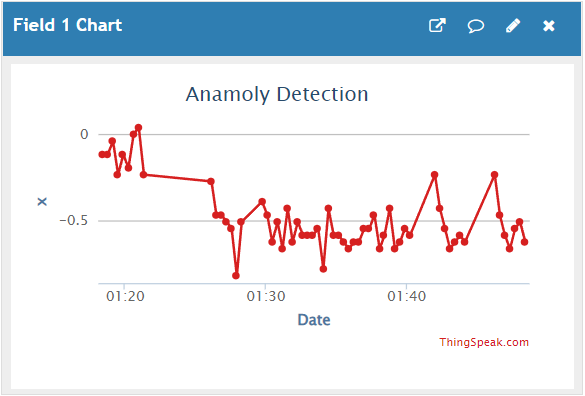


Fig 7. X-axis values

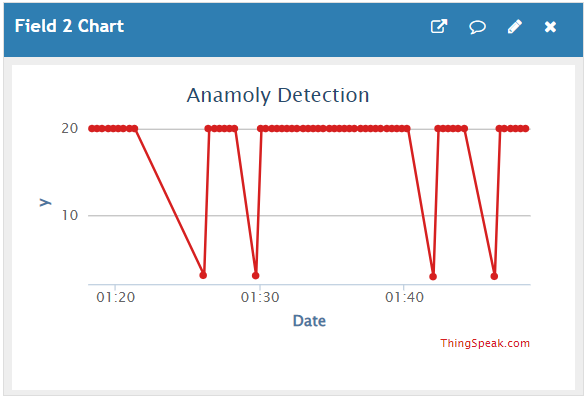


Fig8. Y-axis values

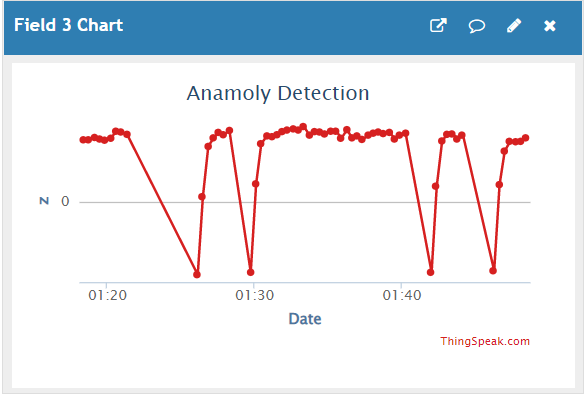


Fig 9 Z-axis values

**Deploying the values through user interface** :

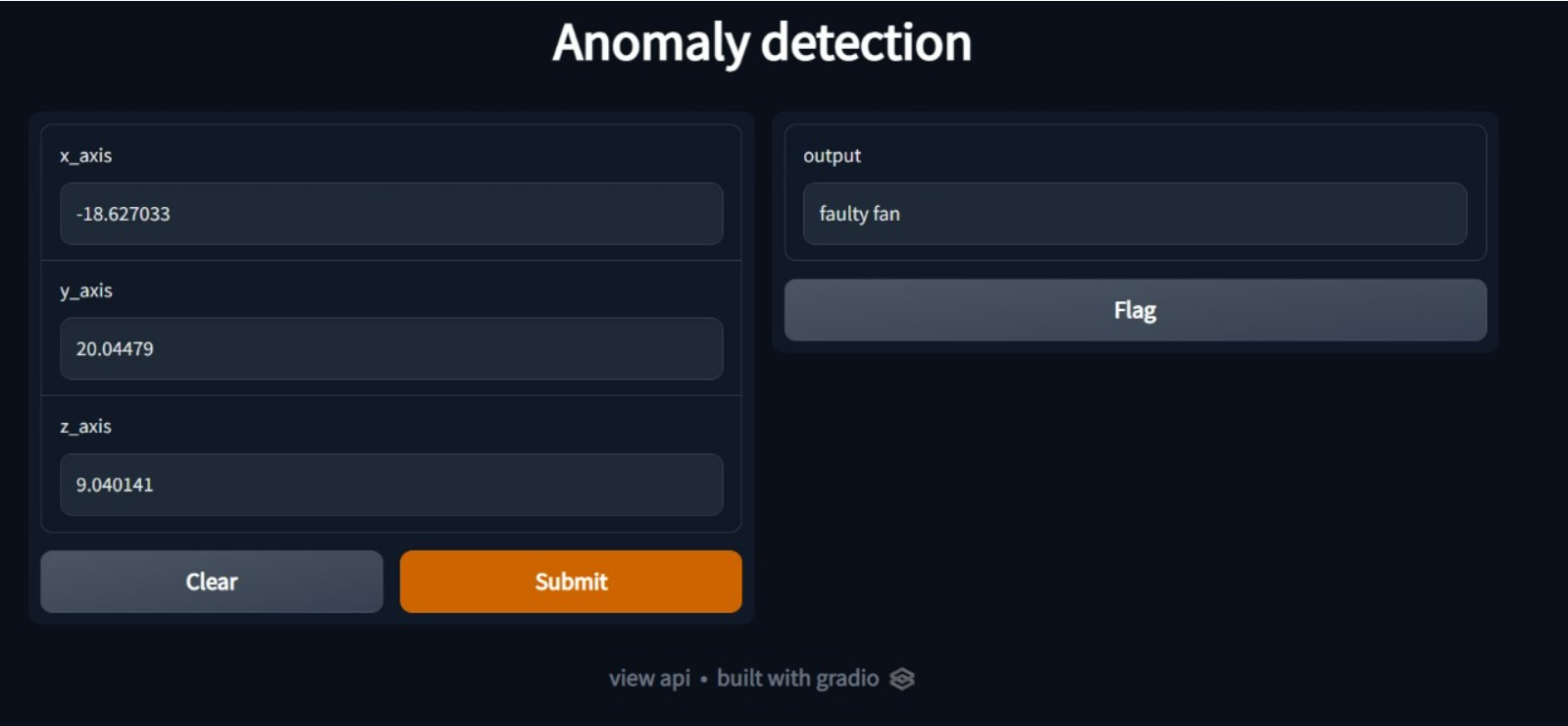


Fig 10 : Classification of a Faulty fan

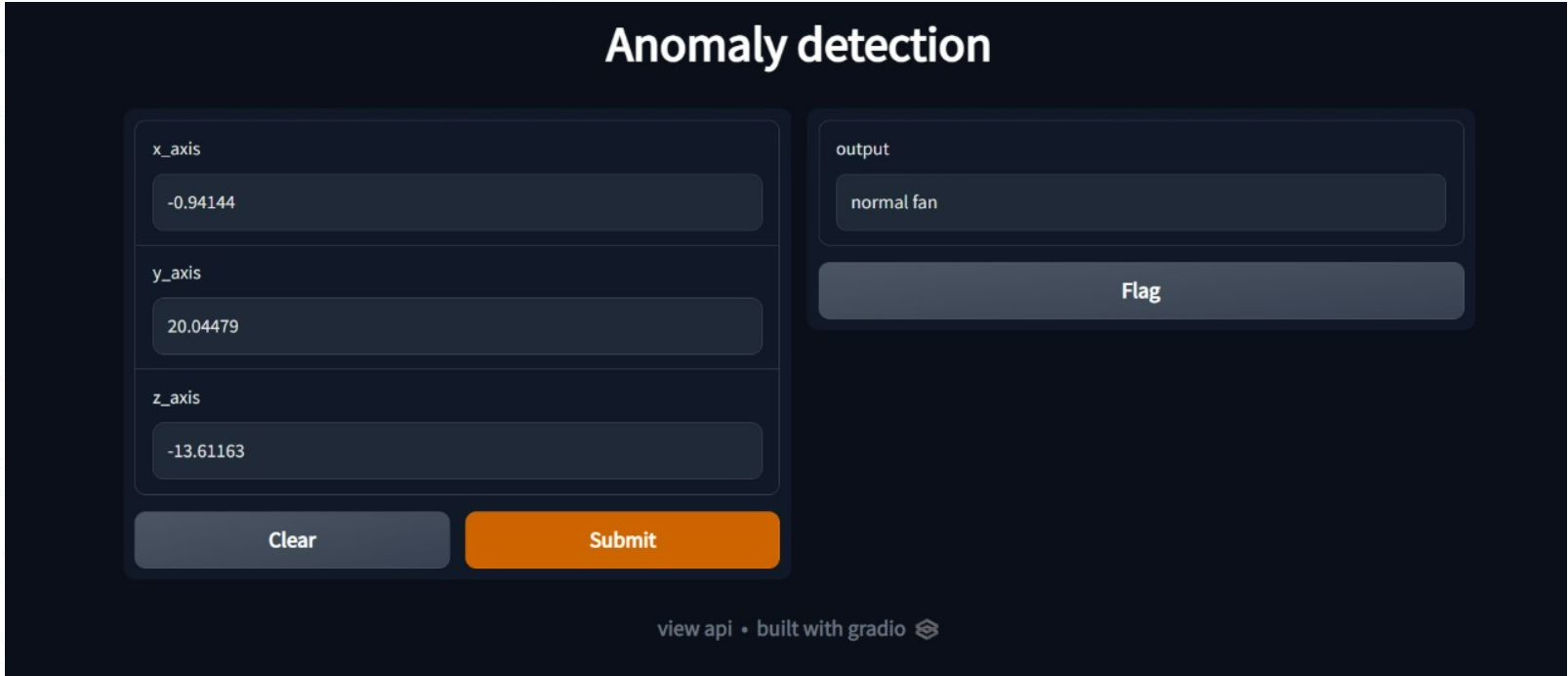


Fig 11 : Classification of a Normal fan

# Future Scope

* In order to build time series models to understand the underlying nature of the time series data or to fit a model for forecasting or monitoring, time series analysis is a statistical technique for studying this time series data. Time series analysis and modelling are crucial for anomaly identification, which involves spotting new or peculiar system states.
* When analysing specific sorts of anomalies over a specific time period or when abnormalities need to be further investigated, the anomaly score approach might be helpful. When various sorts of anomalies are present in robotic finishing machines and these anomalies need to be further studied, for instance, the usage of anomaly score would be suitable.
* The LSTM RNN is used in time series forecasting.

# 6.Conclusion

# The development of the monitoring system for the motor's current condition is the research's main advancement. Data storage, retrieval, and access have been made more user-friendly with the aid of IoT.Vibration-related motor failure issues can simply be communicated to the end user via IoT.Using remote sensing (RS) and IoT technology, this research successfully created and implemented a flexible motor status monitoring system that might decrease downtime for several sectors and industrial businesses.

# 7.Learning Outcomes

* Understand the significance of the Internet of Things for enhancing the convenient safety methods in Smart Factories.
* Planning and discussion of the architecture, operation, and scalability of an IoT solution.
* Interfacing various analog and digital sensors with Node MCU and getting familiar with the Arduino IDE.
* Setting up ThingSpeak channels and retrieving sensor data in the cloud as well as webs browser.

# References

* Review on Anamoly detection in smart manufacturing

<https://arxiv.org/pdf/2107.05053.pdf>

* Reference Paper

[P35\_Demonstration\_of\_In-Network\_Audio\_Processing\_for\_Low-Latency\_Anomaly\_Detection\_in\_Smart\_Factories.pdf](file:///C:\Users\srini\Downloads\P35_Demonstration_of_In-Network_Audio_Processing_for_Low-Latency_Anomaly_Detection_in_Smart_Factories.pdf)

* Reference related to Anomaly Detection

[https://mail.google.com/mail/u/0?ui=2&ik=79dff926c2&attid=0.16&permmsgid=msg- f:1738079384409543247&th=181ee5fdcb29124f&view=att&disp=inline&realattid=f\_l5h1e8la2](https://mail.google.com/mail/u/0?ui=2&ik=79dff926c2&attid=0.16&permmsgid=msg-%20%20%20f:1738079384409543247&th=181ee5fdcb29124f&view=att&disp=inline&realattid=f_l5h1e8la2)

# Source code

#include <Wire.h>

#include <Adafruit\_Sensor.h>

#include <Adafruit\_ADXL345\_U.h>

Adafruit\_ADXL345\_Unified accel = Adafruit\_ADXL345\_Unified();

#include <ESP8266WiFi.h>;

#include <WiFiClient.h>;

#include <ThingSpeak.h>;

const char\* ssid = "POCO M2 Pro"; //Your Network SSID

const char\* password = "dheeraj7092!@#"; //Your Network Password

WiFiClient client;

unsigned long myChannelNumber = 1776533; //Your Channel Number (Without Brackets)

const char \* myWriteAPIKey = "TZQVKCJT5TYPCJVI"; //Your Write API Key

int val;

void setup()

{

Serial.begin(9600);

if(!accel.begin())

{

Serial.println("ADXL not detected");

while(1);

}

delay(10);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED)

{

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

ThingSpeak.begin(client);

}

void loop()

{

float x1=0;

float y=0;

float z=0;

sensors\_event\_t event;

accel.getEvent(&event);

x1=event.acceleration.x;

y=event.acceleration.y;

z=event.acceleration.z;

Serial.print(" ");

Serial.print(event.acceleration.x);

Serial.print(", ");

Serial.print(" ");

Serial.print(event.acceleration.y);

Serial.print(", ");

Serial.print(" ");

Serial.print(event.acceleration.z );

Serial.print(" ");

Serial.println(" ");

ThingSpeak.setField(1, x1);//gas

ThingSpeak.setField(2, y);//button

ThingSpeak.setField(3,z);

// Write to ThingSpeak. There are up to 8 fields in a channel, allowing you to store up to 8 different

// pieces of information in a channel. Here, we write to field 1.

int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);//update data

if(x == 200){//responese ok mean 200

Serial.println("Channel update successful.");

}

else{

Serial.println("Problem updating channel. HTTP error code " + String(x));

}

delay(20000);

}