

CAPSTONE PROJECT -INNOVATE TEMPERATURE AND ANOMALY DETECTION

AIM:

To build a module that detects temperature and any anomaly in the temperature readings and send the mail ,SMS ,alarm through buzzer and LED light.

OBJECTIVE:

This project deals with constantly monitoring temperature using LM35 and BoltIoT Wi-Fi module.

1. To plot a predictor graph made of polynomial regression ML algorithm to predict the future temperature change and ranges so that early action can be taken when the temperature is maintained within the given range for longer than 20 minutes.
2. Sending an Email using Mailgun services and a SMS using Twilio services when temperature is not within the prescribed range so that early action can be taken and when fridge is opened (Using Z-score Analysis i.e. when anomaly in temperature graph is detected) and alarm through the buzzer and LED light.

HARDWARE COMPONENTS:

1. BoltIoT Bolt Wi-Fi Module

Wi-Fi module that gives the hardware connection internet access for transfer of data over cloud.

2. Temperature Sensor LM35 Sensor

3. Jumper wires(generic)

To connect the LM35 to the Wi-Fi module and breadboard.

4. USB-A to Micro-USB Cable

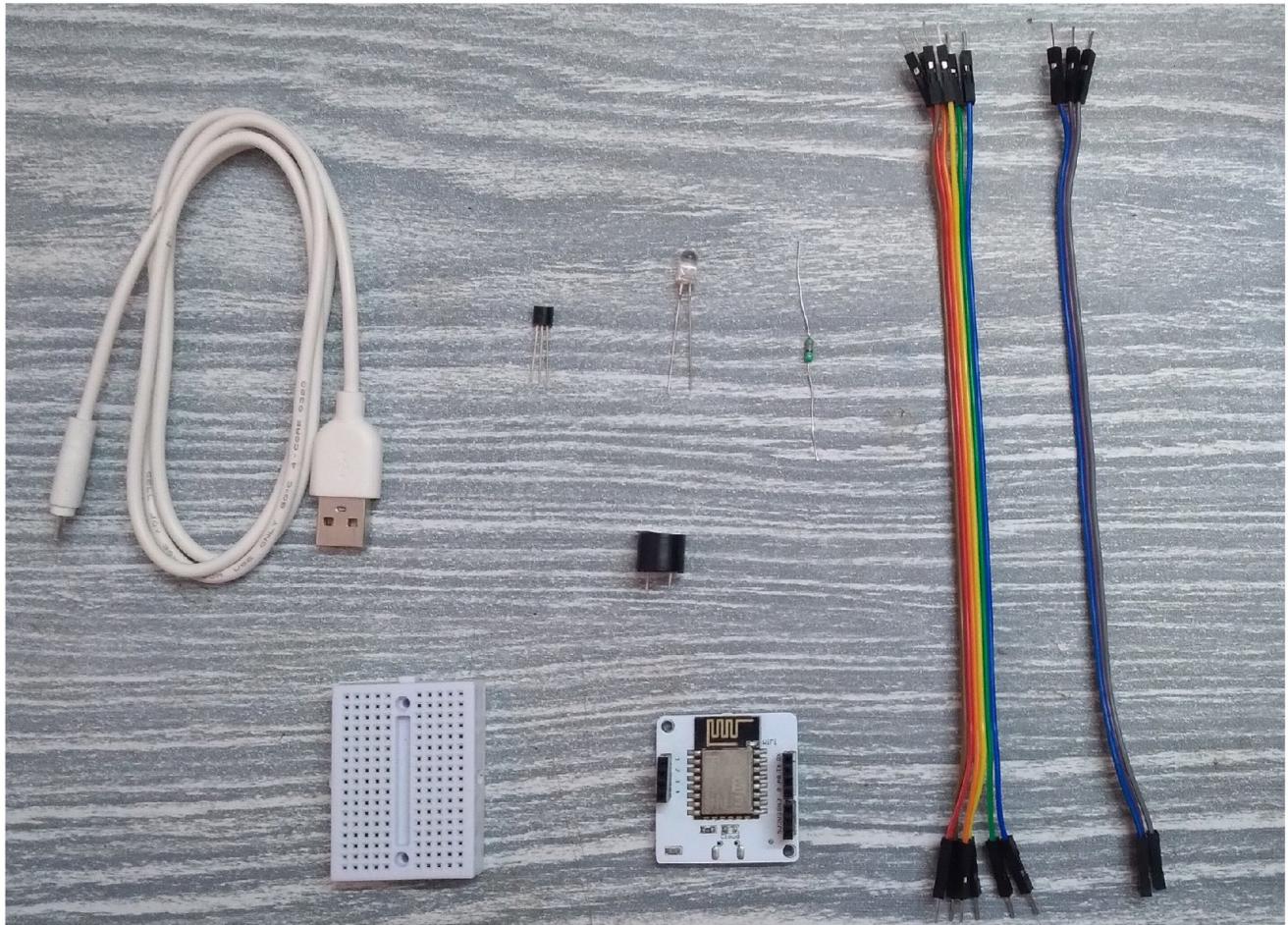
To connect the module to power supply

5. Buzzer

6. Breadboard(generic)

7. LED light

8. Resister of resistance 300 ohm



SOFTWARE APPS AND ONLINE SERVICES

Language used : PYTHON(Machine Learning)

1. Twilio SMS Messaging API

To receive SMS when anomaly(i.e. When fridge door is opened) is detected.

<https://www.twilio.com/>

2. Mailgun

To receive mail when temperature crosses threshold

<https://app.mailgun.com/sessions/new>

3. Bolt IoT Bolt Cloud

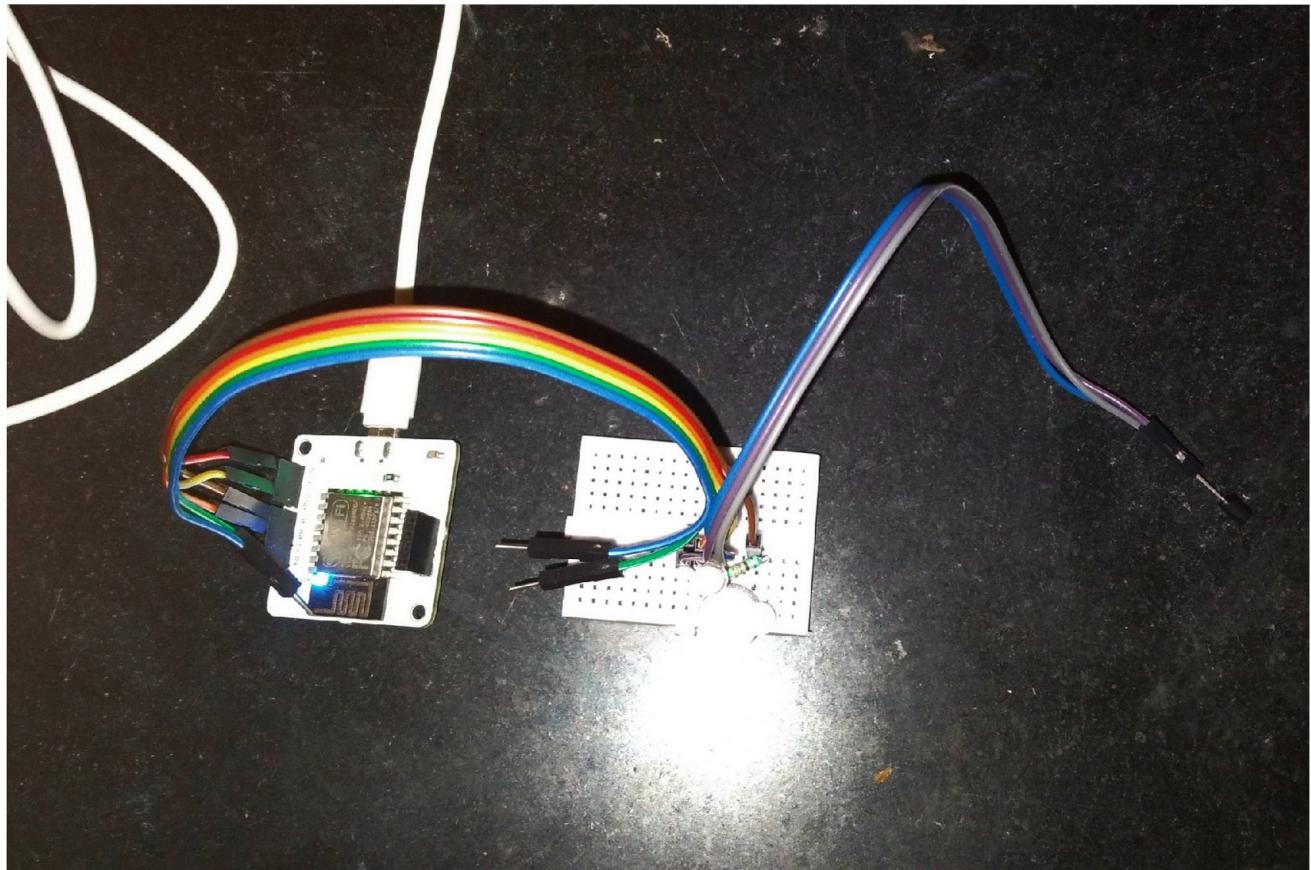
To upload data recorded by the module and process the data

<https://cloud.boltiot.com/>

CONNECTION:

STEPS	COMPONENTS	CONNECTION
1	LM35 to the bolt Module	LM35 pins VCC Bolt module 5v

	using breadboard	Output Gnd	A0(Analog input pin) Gnd
2	Buzzer using breadboard	Buzzer pins Positive end Negative end	Bolt module 0 pin Gnd
3	LED and resistor using breadboard	Led pins Positive end Negative end Another end of resistor	One end of resistor Gnd of Bolt module 0 pin of Bolt module



PROCEDURE:

- 1.Create a product on Bolt cloud to get the upper ,lower temperature limit of the fridge.

Activities Firefox Oct 24 20:30

Fedor Project - Start Page Bolt Cloud cloud.boltiot.com/control

https://cloud.boltiot.com/home/?p_name=Temp_alert

Most Visited Fedora Docs Fedora Magazine Fedora Project User Communities Red Hat Free Content

Products

Devices Products Alerts Assets API Docs New

Light_sensor (0)

led (0)

temp_monitor (0)

Temp_alert (1)

+ Add Product

ID: BOLT13169178 STATUS: ONLINE PRODUCT: Temp_alert ACTIONS: Edit, Delete, More

Help

Activities Firefox Oct 24 20:30

Fedor Project - Start Page Bolt Cloud cloud.boltiot.com/configure/Temp_alert

https://cloud.boltiot.com/configure/Temp_alert

Most Visited Fedora Docs Fedora Magazine Fedora Project User Communities Red Hat Free Content

Products: Setup

Temp.alert Input device • Connected with GPIO

Hardware Code

Feedback

Step 1: Assemble the circuit using Bolt hardware module as per your requirement.
Step 2: Select the pins as per circuit designed and assign a unique variable name to them.
Step 3: Data collection rate: 5 Minutes

Note: Variable name can only contain lowercase alphanumeric characters and underscore and should start with an alphabet.

Pin	Variable Name
A0 Analog	sensor

Help

2. Save a code for polynomial regression and hardware components of product and exit.

POLYNOMIAL CODE

```
setChartLibrary('google-chart');
setChartTitle('Polynomial Regression-Temperature monitor');
setChartType('predictionGraph');
setAxisName('Time', 'Temperature(Celsius)')
mul(0.0977)
setAnimation(true);
setCrosshair(true);
plotChart('time_stamp', 'sensor');
```

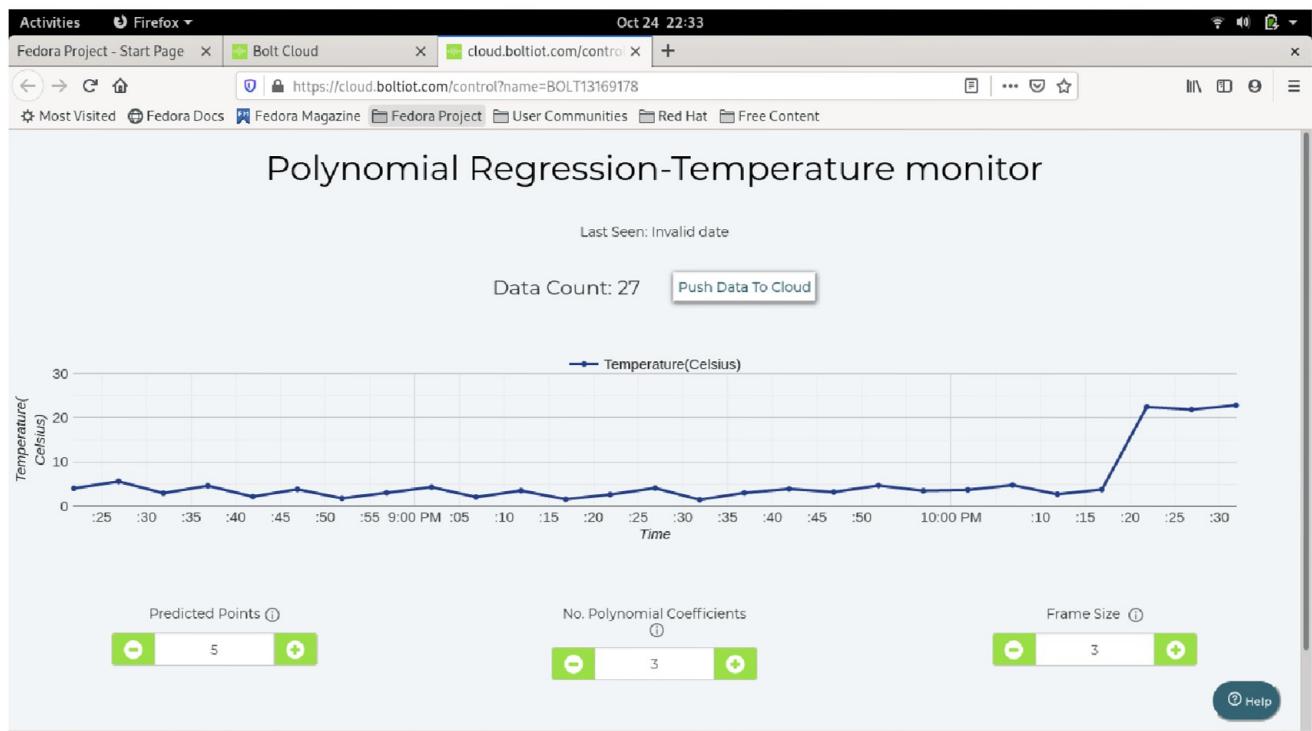
3. Later link the product to the bolt module and deploy the configuration.

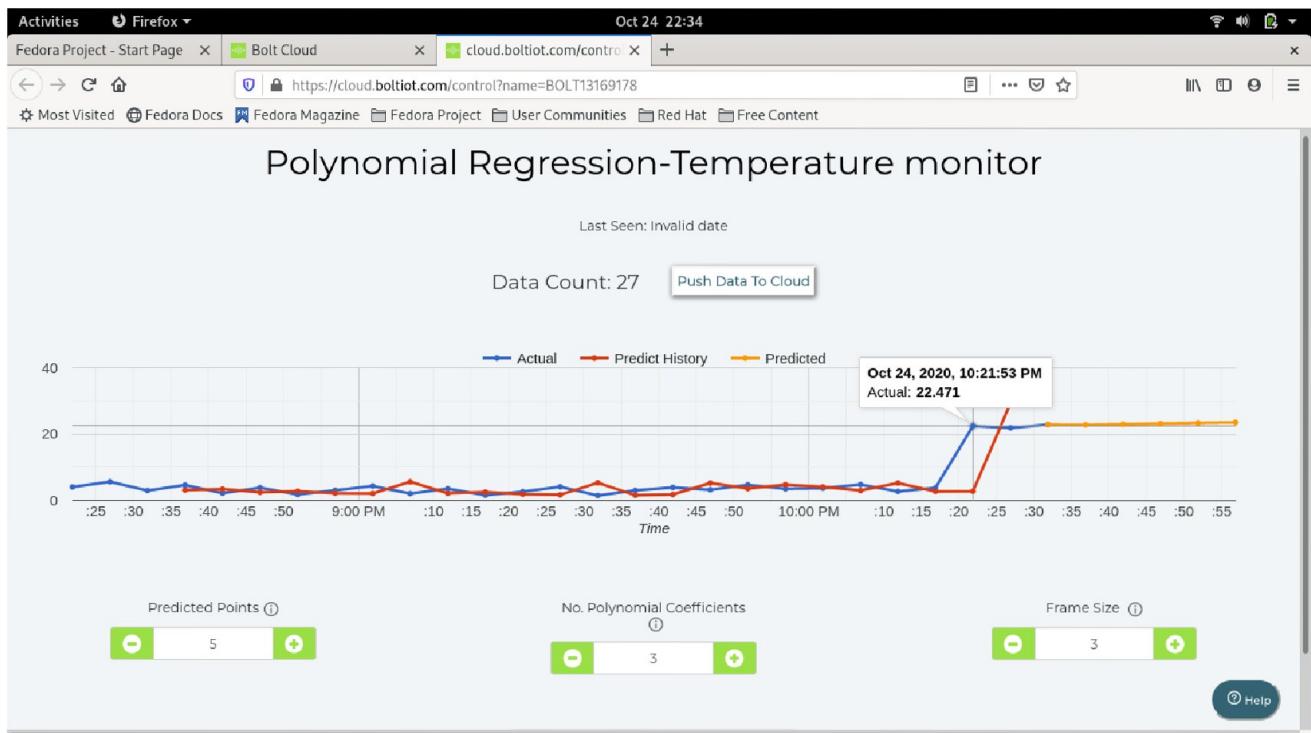
The screenshot shows the Bolt Cloud interface in a Firefox browser window. The title bar indicates it's Oct 24 20:31. The main area is titled "Products: Setup" under "Temp_alert". A sidebar on the left shows "Input device connected with GPIO". The "Code" tab is selected, displaying a code editor with the following content:

```
temp_alert
is
1 setChartLibrary('google-chart');
2 setChartTitle('Polynomial Regression-Temperature monitor');
3 setChartType('predictionGraph');
4 setAxisName('Time','Temperature(Celsius)');
5 mul(0.0977)
6 setAnimation(true);
7 setCrosshair(true);
8 plotChart('time_stamp','sensor');
```

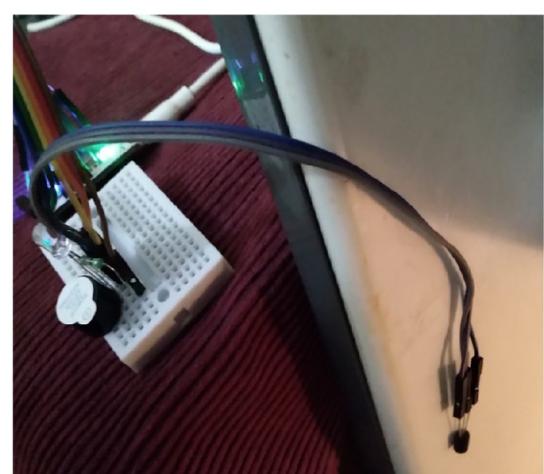
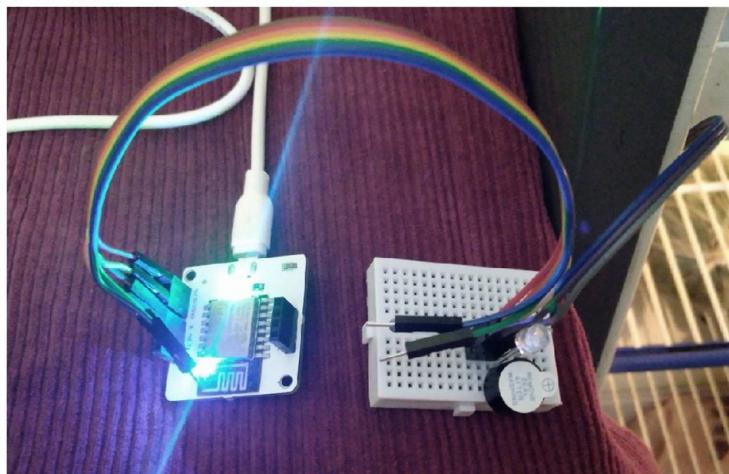
Below the code editor, there's a note: "Note : Variable name can only contain lowercase alphanumeric characters and underscore and should start with an alphabet." A table titled "Pin" and "Variable Name" shows one entry: "A0 Analog" mapped to "sensor". A "Feedback" button is visible on the left, and a "Help" button is at the bottom right.

PREDICTION GRAPH





4. Place the temperature sensor for about 2 hours inside the fridge and see the output.



5.Analyse the output and decide the maximum and minimum limit.

Min=2 degree= 20.48 (sensor value)

Max=4.5 degree= 46.08(sensor value)

6. Install BoltIot python library and write python code on the terminal in the linux operating system.
(or digital ocean or virtual machine)

7.Create a new folder temp_alert by following command.

```
mkdir temp_alert  
cd temp_alert
```

```
sudo atom conf.py
```

Note : Need a malignun account and twilio SMS messaging account before performing the upcoming steps.

8.Create a file named ‘conf.py’ in folder ‘temp_alert’ which will store all the credentials related to malignun ,twilio ,bolt device id ,bolt API key,mul_factor and frame size.

CONF . PY

```
SID='You can find SID in your Twilio Dashboard'  
AUTH_TOKEN='You can find on your Twilio Dashboard'  
FROM_NUMBER='This is the no. generated by Twilio. You can find this on your  
Twilio Dashboard'  
TO_NUMBER='This is your number. Make sure you are adding +91 in beginning'  
API_KEY='This is your Bolt Cloud accout API key'  
DEVICE_ID='This is the ID of your Bolt device'  
MAILGUN_API_KEY='This is the private API key which you can find on your Malignun  
Dashboard'  
SANDBOX_URL='You can find this on your Malignun Dashboard'  
RECIPIENT_EMAIL='Enter your Email ID Here'  
SENDER_EMAIL='This would be test@your SANDBOX_URL'  
FRAME_SIZE=10  
MUL_FACTOR=6
```

9.Create one more file ‘main.py’ in folder ‘temp_alert’ using command

```
sudo nano main.py
```

10.ALGORITHM OF THE PYTHON CODE:

1. Import conf which have all credentials ,json,time.

```
import conf,json,time,math,statistics
```

2. Import our Bolt python library which will let us fetch the data stored in bolt Cloud and then based on it send mail and sms and import math and statics libraries will be required for calculating the z-score and the threshold boundaries.

```
from boltiot import Sms,Bolt,Email  
history_data=[]  
min=20.48  
max=46.08  
mybolt=Bolt(conf.API_KEY,conf.DEVICE_ID)  
sms=Sms(conf.SID,conf.AUTH_TOKEN,conf.TO_NUMBER,conf.FROM_NUMBER)
```

```
mailer=Email(conf.MAILGUN_API_KEY,conf.SANDBOX_URL,conf.SENDER_EMAIL,conf.RECEPIENT_EMAIL)
```

3. Fetch the sensor value from bolt service and store in a list ‘history_data’ which is used for computing z-score.
4. Display the sensor value in terms of temperature (celcius).

Temperature(celcius)=Sensor value/10.24

```
print("Reading sensor value")
response=mybolt.analogRead('A0')
data=json.loads(response)
if data['success']!=1:
    print("There was an error while retrieving the data.")
    print("This is the error:"+data['value'])
    time.sleep(10)
    continue
try:
    sensor_value=int(data['value'])
except Exception as e:
    print("There was an error while parsing the response:",e)
    continue
value=sensor_value/10.24
print("THE CURRENT TEMPERATURE:"+str(value))
```

5. Define a function ‘compute’ to compute z-score ,upper and lower threshold bounds for normal and anomalous readings.

```
def compute(history_data,frame_size,factor):
    if len(history_data)<frame_size:
        return None
    if len(history_data)>frame_size:
        del history_data[0:len(history_data)-frame_size]
        Mn=statistics.mean(history_data)
        variance=0
        for data in history_data:
            variance+=math.pow((data-Mn),2)
        Zn=factor*math.sqrt(variance/frame_size)
        High_bound=history_data[frame_size-1]+Zn
        Low_bound=history_data[frame_size-1]-Zn
        return [High_bound,Low_bound]
```

6. If the sensor reading cross the range of normal reading send a mail and a SMS , if any error occurred display the error.

```
try:
    if sensor_value>max or sensor_value<min:
        mybolt.digitalWrite('0','HIGH')
        print("TEMPERATURE CROSSED THE LIMIT")
        response1=sms.send_sms("TEMPERATURE CROSSED THE LIMIT.THE CURRENT
TEMPERATURE IS "+str(value))
        response2=mailer.send_email("ALERT!","TEMPERATURE CROSSED THE
LIMIT.THE CURRENT TEMPERATURE IS "+str(value))
        print(response2)
        print("Status of SMS at Twilio is:"+str(response1.status))
        response_text=json.loads(response2.text)
        print("Response received from Mailgun
is:"+str(response_text['message']))
        time.sleep(10)
        mybolt.digitalWrite('0','LOW')
except Exception as e:
```

```

    print("Error occurred:Below are the details")
    print(e)
    time.sleep(10)
    continue

```

7. Then depending on bounds of Z-score analysis check the sensor value ,if it crossed the limit display ‘Someone opened the door’ and send a mail and a SMS,if any error occurred display the error.

```

bound=compute(history_data,conf.FRAME_SIZE,conf.MUL_FACTOR)
if not bound:
    count=conf.FRAME_SIZE-len(history_data)
    print("Not enough data to compute Z-score.Need",count,"more data points")
    history_data.append(int(data['value']))
    time.sleep(10)
    continue
try:
    if sensor_value> bound[0] or sensor_value<bound[1]:
        mybolt.digitalWrite('0','HIGH')
        print("Alert! Someone opened the door")
        response1=sms.send_sms("ALERT! SOMEONE OPENED THE DOOR")
        print("Status of SMS at Twilio is:"+str(response1.status))
        response2=mailer.send_email("ALERT!","SOMEONE OPENED THE DOOR")
        response_text=json.loads(response2.text)
        print("This is the response for mail "+str(response_text['message']))
        time.sleep(10)
        mybolt.digitalWrite('0','LOW')
        history_data.append(sensor_value)
    except Exception as e:
        print("Error",e)

```

8. Wait for 10 seconds.

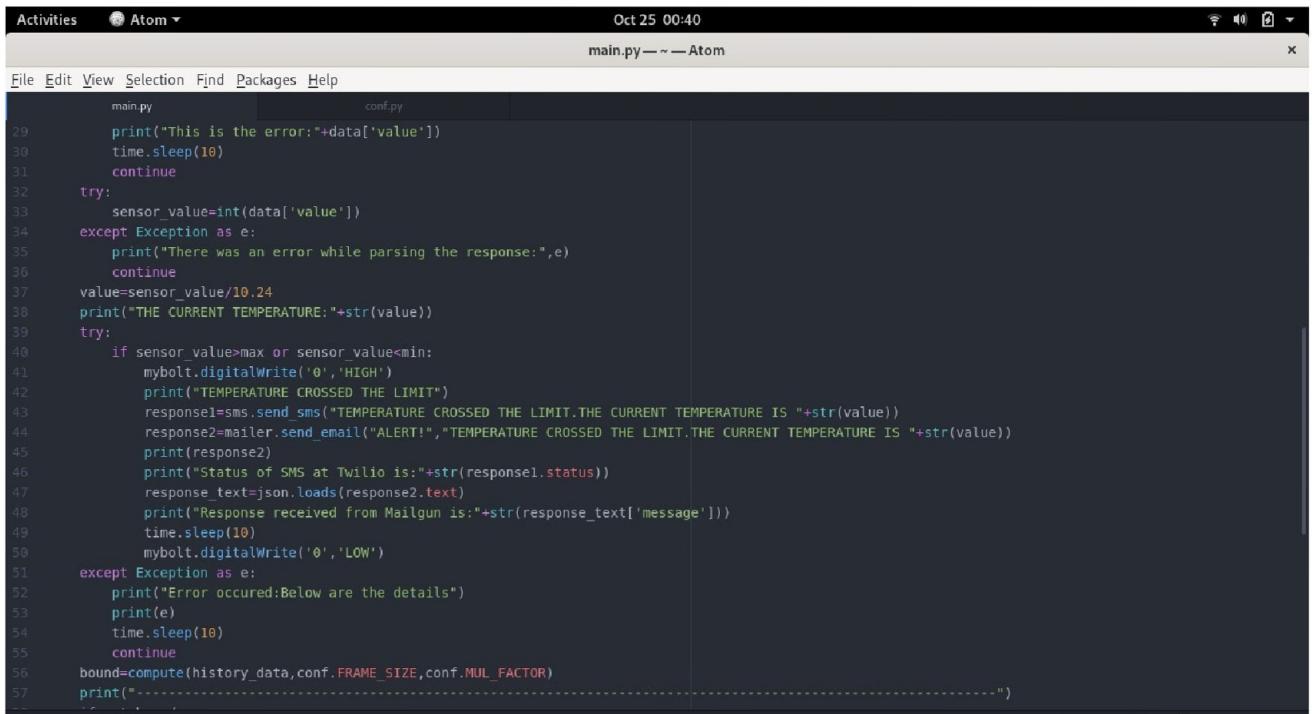
```
time.sleep(10)
```

9. Repeat the steps from step3.

```

Activities Atom Oct 25 00:40
main.py — ~ — Atom
File Edit View Selection Find Packages Help
main.py | conf.py x
Oct 25 00:40
main.py — ~ — Atom
1 import conf,json,time,math,statistics
2 from boltiot import Sms,Bolt,Email
3 def compute(history_data,frame_size,factor):
4     if len(history_data)<frame_size:
5         return None
6     if len(history_data)>frame_size:
7         del history_data[0:len(history_data)-frame_size]
8         Mn=statistics.mean(history_data)
9         variance=0
10        for data in history_data:
11            variance+=math.pow((data-Mn),2)
12        Zn=factor*math.sqrt(variance/frame_size)
13        High_bound=history_data[frame_size-1]+Zn
14        Low_bound=history_data[frame_size-1]-Zn
15        return [High_bound,Low_bound]
16 history_data=[]
17 min=20.48
18 max=46.08
19 mybolt=Bolt(conf.API_KEY,conf.DEVICE_ID)
20 sms=Sms(conf.SID,conf.AUTH_TOKEN,conf.TO_NUMBER,conf.FROM_NUMBER)
21 mailer=Email(conf.MAILGUN_API_KEY,conf.SANDBOX_URL,conf.SENDER_EMAIL,conf.RECIPIENT_EMAIL)
22 while True:
23     print("====")
24     print("Reading sensor value")
25     response=mybolt.analogRead('A0')
26     data=json.loads(response)
27     if data['success']==1:
28         print("There was an error while retriving the data.")
29         print("This is the error:"+data['value'])
30         time.sleep(10)
temp_alert/main.py 72:59

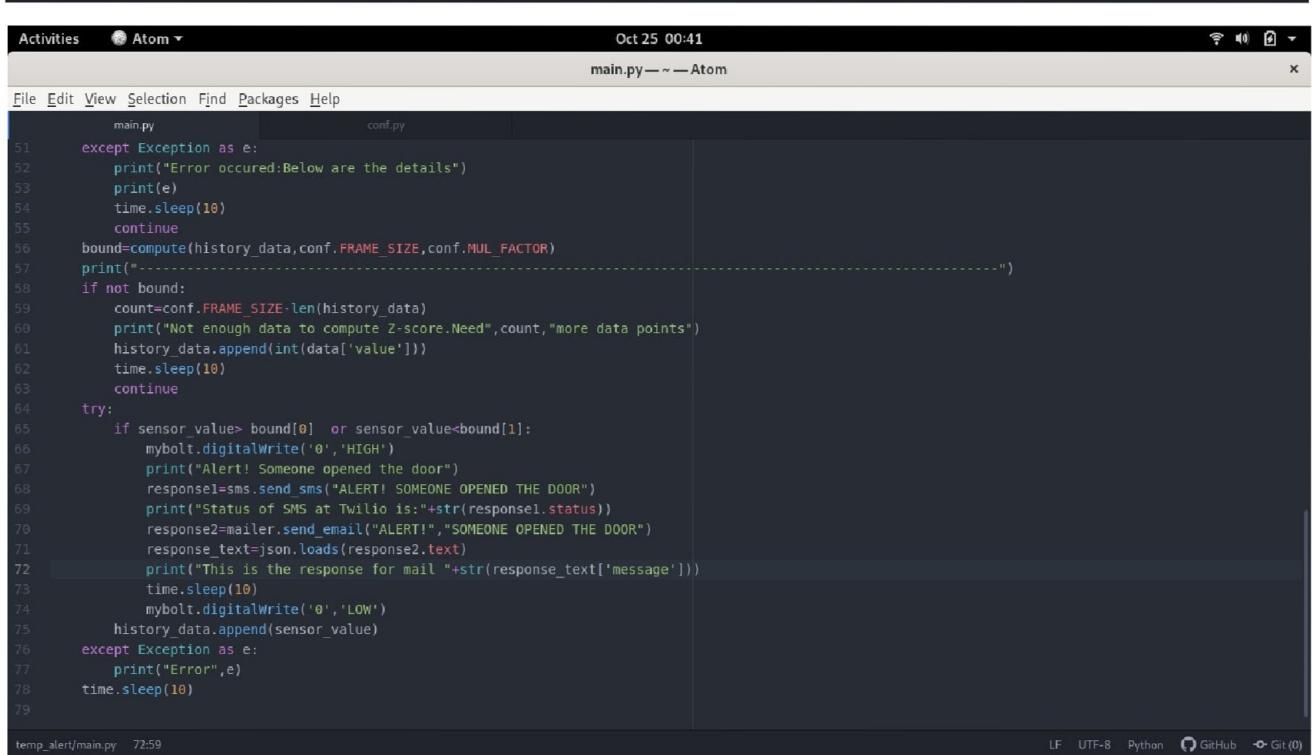
```



```

Oct 25 00:40
main.py — ~ — Atom
File Edit View Selection Find Packages Help
main.py | conf.py
29     print("This is the error:"+data['value'])
30     time.sleep(10)
31     continue
32   try:
33     sensor_value=int(data['value'])
34   except Exception as e:
35     print("There was an error while parsing the response:",e)
36     continue
37   value=sensor_value/10.24
38   print("THE CURRENT TEMPERATURE:"+str(value))
39   try:
40     if sensor_value>max or sensor_value<min:
41       mybolt.digitalWrite('0','HIGH')
42       print("TEMPERATURE CROSSED THE LIMIT")
43       response1=sms.send_sms("TEMPERATURE CROSSED THE LIMIT.THE CURRENT TEMPERATURE IS "+str(value))
44       response2=mailer.send_email("ALERT!","TEMPERATURE CROSSED THE LIMIT.THE CURRENT TEMPERATURE IS "+str(value))
45       print(response2)
46       print("Status of SMS at Twilio is:"+str(response1.status))
47       response_text=json.loads(response2.text)
48       print("Response received from Mailgun is:"+str(response_text['message']))
49       time.sleep(10)
50       mybolt.digitalWrite('0','LOW')
51   except Exception as e:
52     print("Error occurred:Below are the details")
53     print(e)
54     time.sleep(10)
55     continue
56   bound=compute(history_data,conf.FRAME_SIZE,conf.MUL_FACTOR)
57   print("-----")
temp_alert/main.py 72:59
LF  UTF-8  Python  GitHub  Git (0)

```

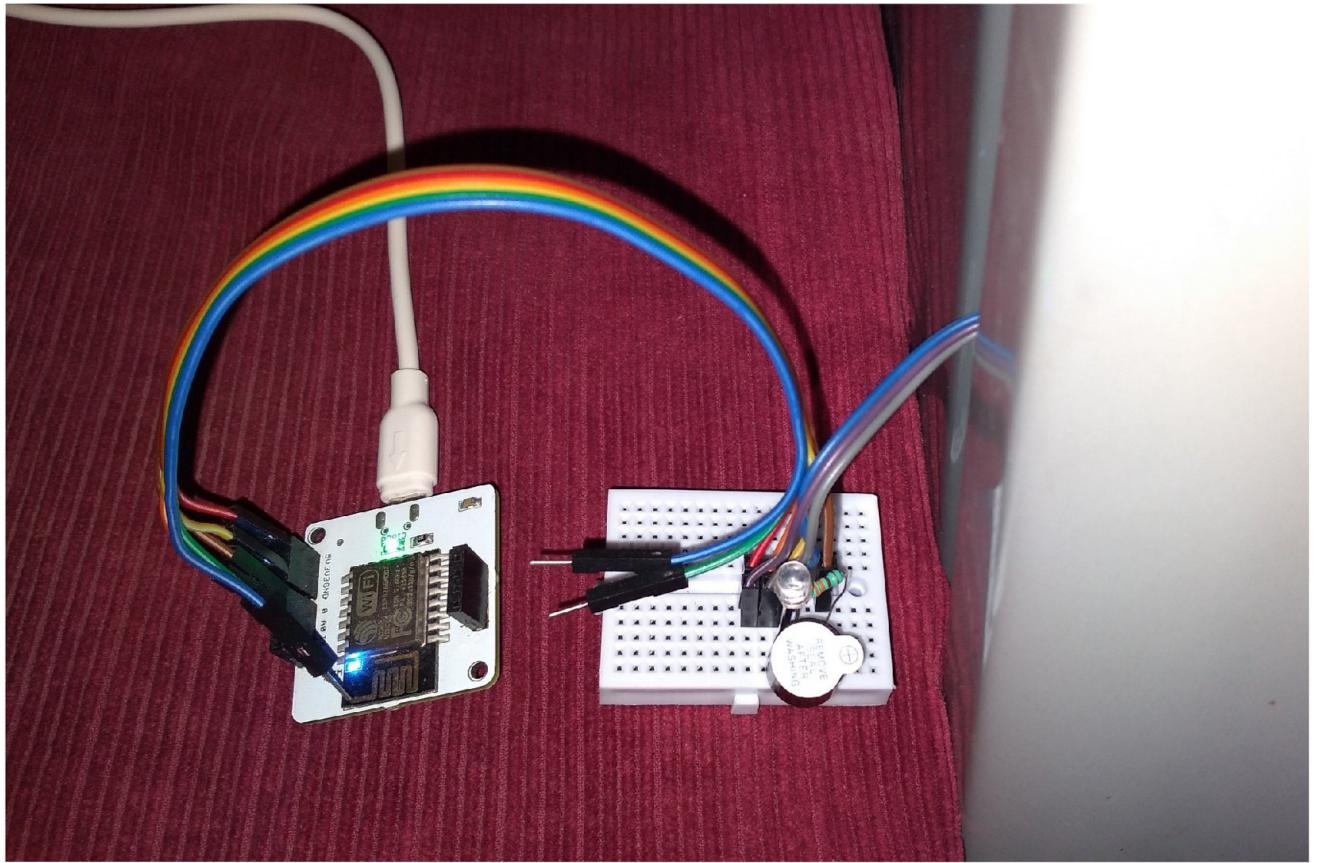
```

Oct 25 00:41
main.py — ~ — Atom
File Edit View Selection Find Packages Help
main.py | conf.py
51   except Exception as e:
52     print("Error occurred:Below are the details")
53     print(e)
54     time.sleep(10)
55     continue
56   bound=compute(history_data,conf.FRAME_SIZE,conf.MUL_FACTOR)
57   print("-----")
58   if not bound:
59     count=conf.FRAME_SIZE-len(history_data)
60     print("Not enough data to compute Z-score.Need",count,"more data points")
61     history_data.append(int(data['value']))
62     time.sleep(10)
63     continue
64   try:
65     if sensor_value> bound[0]  or sensor_value<bound[1]:
66       mybolt.digitalWrite('0', 'HIGH')
67       print("Alert! Someone opened the door")
68       response1=sms.send_sms("ALERT! SOMEONE OPENED THE DOOR")
69       print("Status of SMS at Twilio is:"+str(response1.status))
70       response2=mailer.send_email("ALERT!","SOMEONE OPENED THE DOOR")
71       response_text=json.loads(response2.text)
72       print("This is the response for mail "+str(response_text['message']))
73       time.sleep(10)
74       mybolt.digitalWrite('0','LOW')
75       history_data.append(sensor_value)
76   except Exception as e:
77     print("Error",e)
78     time.sleep(10)
79
temp_alert/main.py 72:59
LF  UTF-8  Python  GitHub  Git (0)

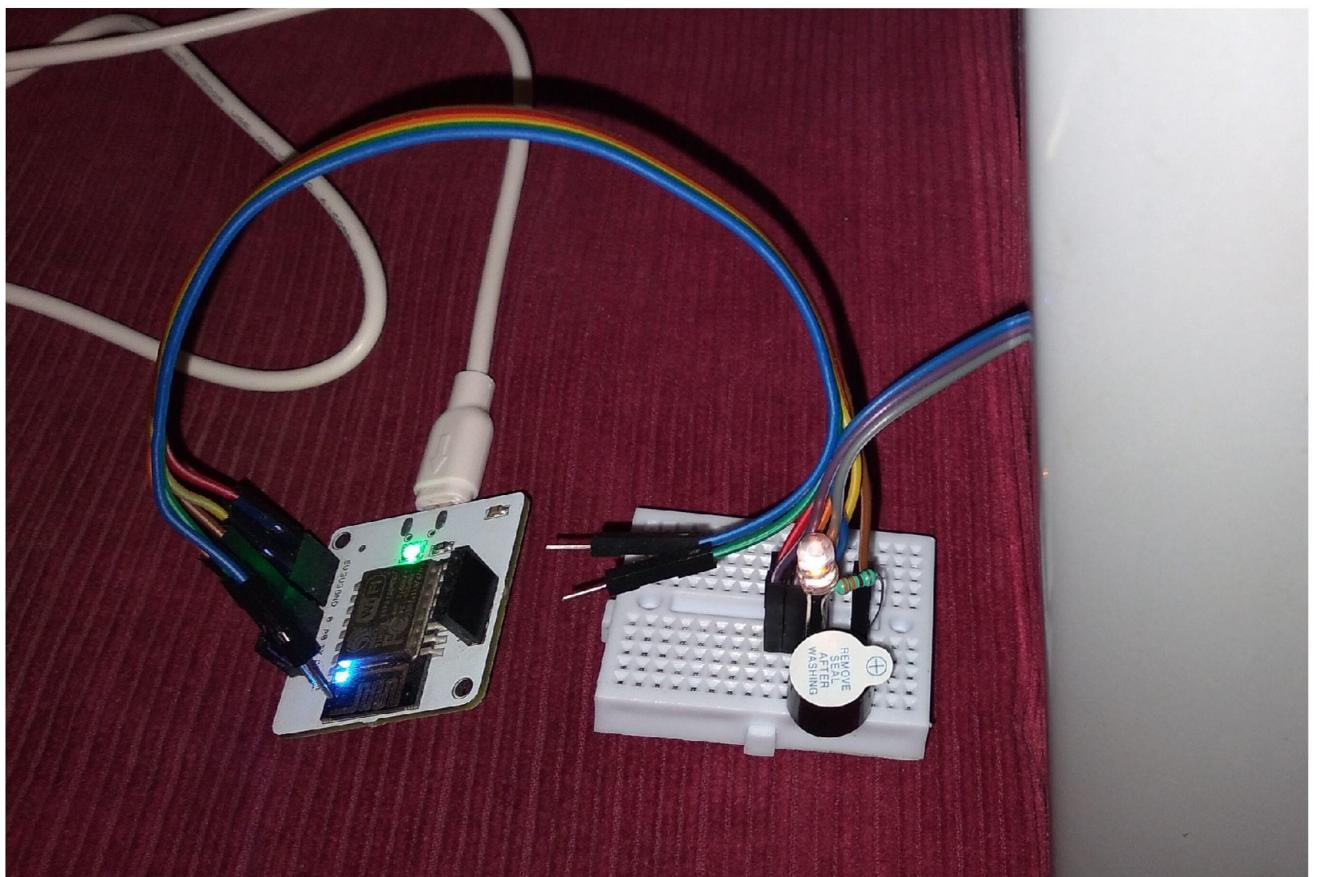
```

11.Run the python code by following code.

`sudo python3 main.py`



When anomaly or temperature variation from range found, LED glows and buzzer alarms.



OUTPUT:

```
Activities Terminal Oct 25 00:38
GanapriyaS@localhost:~/temp_alert — sudo python3 main.py

[GanapriyaS@localhost temp_alert]$ sudo python3 main.py
[sudo] password for GanapriyaS:
=====
Reading sensor value
THE CURRENT TEMPERATURE:4.296875

Not enough data to compute Z-score.Need 10 more data points
=====

Reading sensor value
THE CURRENT TEMPERATURE:4.296875

Not enough data to compute Z-score.Need 9 more data points
=====

Reading sensor value
THE CURRENT TEMPERATURE:4.296875

Not enough data to compute Z-score.Need 8 more data points
=====

Reading sensor value
THE CURRENT TEMPERATURE:4.296875

Not enough data to compute Z-score.Need 7 more data points
=====

Reading sensor value
THE CURRENT TEMPERATURE:4.296875

Not enough data to compute Z-score.Need 6 more data points
=====

Reading sensor value
THE CURRENT TEMPERATURE:4.19921875

Not enough data to compute Z-score.Need 5 more data points
=====

Reading sensor value
THE CURRENT TEMPERATURE:4.296875

Not enough data to compute Z-score.Need 4 more data points
=====

Reading sensor value
```

```
Activities Terminal Oct 25 00:38
GanapriyaS@localhost:~/temp_alert — sudo python3 main.py

Reading sensor value
THE CURRENT TEMPERATURE:4.296875

Not enough data to compute Z-score.Need 6 more data points
=====

Reading sensor value
THE CURRENT TEMPERATURE:4.19921875

Not enough data to compute Z-score.Need 5 more data points
=====

Reading sensor value
THE CURRENT TEMPERATURE:4.296875

Not enough data to compute Z-score.Need 4 more data points
=====

Reading sensor value
THE CURRENT TEMPERATURE:4.1015625

Not enough data to compute Z-score.Need 3 more data points
=====

Reading sensor value
THE CURRENT TEMPERATURE:4.19921875

Not enough data to compute Z-score.Need 2 more data points
=====

Reading sensor value
THE CURRENT TEMPERATURE:4.1015625

Not enough data to compute Z-score.Need 1 more data points
=====

Reading sensor value
THE CURRENT TEMPERATURE:4.1015625

Not enough data to compute Z-score.Need 0 more data points
=====

Reading sensor value
THE CURRENT TEMPERATURE:4.19921875
```

```
Activities Terminal ▾ Oct 25 00:39
GanapriyaS@localhost:~/temp_alert — sudo python3 main.py

=====
Reading sensor value
THE CURRENT TEMPERATURE:4.1015625
=====

=====
Reading sensor value
THE CURRENT TEMPERATURE:4.1015625
=====

=====
Reading sensor value
THE CURRENT TEMPERATURE:9.765625
TEMPERATURE CROSSED THE LIMIT
<Response [200]>
Status of SMS at Twilio is:queued
Response received from Mailgun is:Queued. Thank you.

=====
Alert! Someone opened the door
Status of SMS at Twilio is:queued
This is the response for mail Queued. Thank you.

=====
Reading sensor value
THE CURRENT TEMPERATURE:20.3125
TEMPERATURE CROSSED THE LIMIT
<Response [200]>
Status of SMS at Twilio is:queued
Response received from Mailgun is:Queued. Thank you.

=====
Alert! Someone opened the door
Status of SMS at Twilio is:queued
This is the response for mail Queued. Thank you.
```

Activities Firefox ▾ Oct 25 14:42

Fedora Project - Start Page M ALERT! - ganapriyakheers +

https://mail.google.com/mail/u/0/#spam/FMfcgxwKjBMchWfwSwWGXqTgllznlSC

Most Visited Fedora Docs Fedora Magazine Fedora Project User Communities Red Hat Free Content

Gmail in:spam 1 of 1

ALERT! Spam

test@sandbox42f7641e9b7d41808aa7e0a675ed41e9.mailgun.org 00:01 (14 hours ago) ☆
TEMPERATURE CROSSED THE LIMIT.THE CURRENT TEMPERATURE IS 9.765625

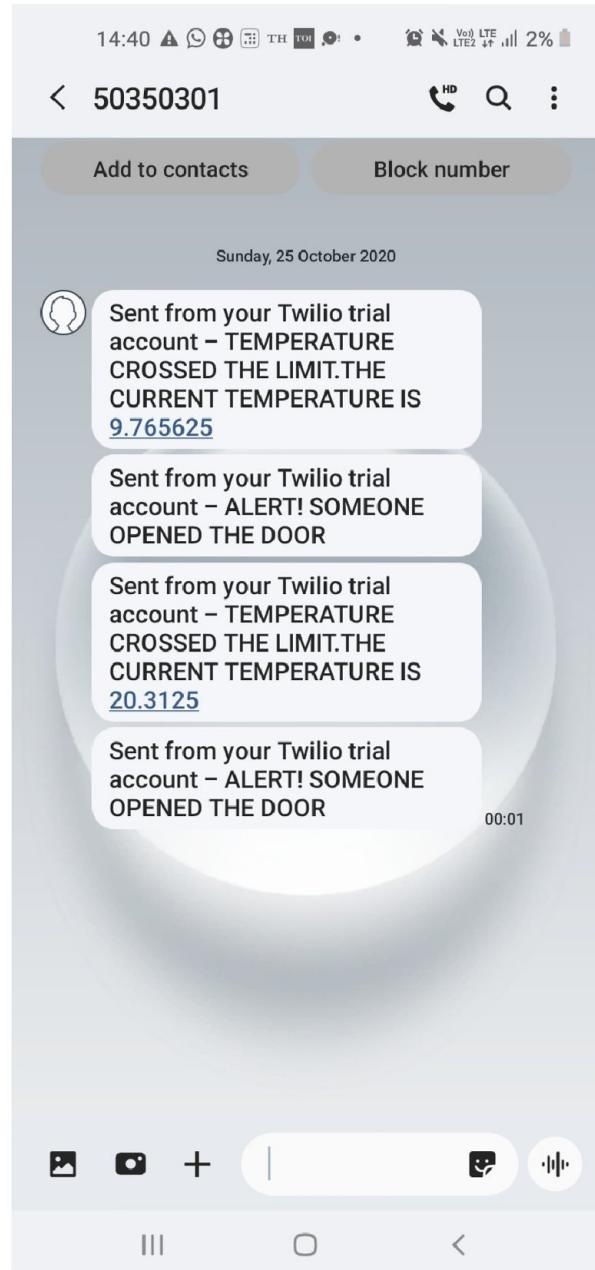
test@sandbox42f7641e9b7d41808aa7e0a675ed41e9.mailgun.org 00:01 (14 hours ago) ☆
TEMPERATURE CROSSED THE LIMIT.THE CURRENT TEMPERATURE IS 20.3125

test@sandbox42f7641e9b7d41808aa7e0a675ed41e9.mailgun.org 00:01 (14 hours ago) ☆
SOMEONE OPENED THE DOOR

test@sandbox42f7641e9b7d41808aa7e0a675ed41e9.mailgun.org via sandbox.mgsend.net 00:01 (14 hours ago) ☆ ↗
to me
SOMEONE OPENED THE DOOR

Reply Forward

No H 31 deleted messages in this conversation View messages or delete forever



COMPLETE PROGRAM CODE

```
import conf,json,time,math,statistics
from boltiot import Sms,Bolt,Email
def compute(history_data,frame_size,factor):
    if len(history_data)<frame_size:
        return None
    if len(history_data)>frame_size:
        del history_data[0:len(history_data)-frame_size]
        Mn=statistics.mean(history_data)
        variance=0
        for data in history_data:
            variance+=math.pow((data-Mn),2)
        Zn=factor*math.sqrt(variance/frame_size)
        High_bound=history_data[frame_size-1]+Zn
        Low_bound=history_data[frame_size-1]-Zn
        return [High_bound,Low_bound]
```

```

history_data=[]
min=20.48
max=46.08
mybolt=Bolt(conf.API_KEY,conf.DEVICE_ID)
sms=Sms(conf.SID,conf.AUTH_TOKEN,conf.TO_NUMBER,conf.FROM_NUMBER)
mailer=Email(conf.MAILGUN_API_KEY,conf.SANDBOX_URL,conf.SENDER_EMAIL,conf.RECEPIE
NT_EMAIL)
while True:

    print("====")
    print("Reading sensor value")
    response=mybolt.analogRead('A0')
    data=json.loads(response)
    if data['success']!=1:
        print("There was an error while retriving the data.")
        print("This is the error:"+data['value'])
        time.sleep(10)
        continue
    try:
        sensor_value=int(data['value'])
    except Exception as e:
        print("There was an error while parsing the response:",e)
        continue
    value=sensor_value/10.24
    print("THE CURRENT TEMPERATURE:"+str(value))
    try:
        if sensor_value>max or sensor_value<min:
            mybolt.digitalWrite('0','HIGH')
            print("TEMPERATURE CROSSED THE LIMIT")
            response1=sms.send_sms("TEMPERATURE CROSSED THE LIMIT.THE CURRENT
TEMPERATURE IS "+str(value))
            response2=mailer.send_email("ALERT!","TEMPERATURE CROSSED THE
LIMIT.THE CURRENT TEMPERATURE IS "+str(value))
            print(response2)
            print("Status of SMS at Twilio is:"+str(response1.status))
            response_text=json.loads(response2.text)
            print("Response received from Mailgun
is:"+str(response_text['message']))
            time.sleep(10)
            mybolt.digitalWrite('0','LOW')
    except Exception as e:
        print("Error occured:Below are the details")
        print(e)
        time.sleep(10)
        continue
    bound=compute(history_data,conf.FRAME_SIZE,conf.MUL_FACTOR)

    print("-----")
    if not bound:
        count=conf.FRAME_SIZE-len(history_data)
        print("Not enough data to compute Z-score.Need",count,"more data points")
        history_data.append(int(data['value']))
        time.sleep(10)
        continue
    try:
        if sensor_value> bound[0] or sensor_value<bound[1]:
            mybolt.digitalWrite('0','HIGH')
            print("Alert! Someone opened the door")
            response1=sms.send_sms("ALERT! SOMEONE OPENED THE DOOR")
            print("Status of SMS at Twilio is:"+str(response1.status))
            response2=mailer.send_email("ALERT!","SOMEONE OPENED THE DOOR")
            response_text=json.loads(response2.text)

```

```
    print("This is the response for mail "+str(response_text['message']))
    time.sleep(10)
    mybolt.digitalWrite('0','LOW')
    history_data.append(sensor_value)
except Exception as e:
    print("Error",e)
time.sleep(10)
```

RESULT:

The final output of prediction chart in bolt cloud gives the prediction of next 20 minutes of temperature range. And the python code output detect and alert the change in temperature and an anomaly in the temperature when the door is opened by mail ,SMS and alarm by LED light and buzzer. As the result of this project pharmaceutical companies keep their tablets in better condition to avoid destroying ,major loss and can able to obey all the government rules perfectly.