iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** IX **Month of publication:** September 2022

DOI: https://doi.org/10.22214/ijraset.2022.46844

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue IX Sep 2022- Available at www.ijraset.com

Anomaly Detection in Ambient Air Quality

Sneha Samuel
Student of Bachelors in Electronics Engineering at University of Mumbai, Maharashtra , India.

Abstract: ***In smart cities there are systems which monitor the air pollution levels at different locations. Supervising the information given by the gas sensor devices may be a tedious task because the sensor data is simply too huge to investigate manually. Further because of electrical interference and environmental conditions, the sensors may show false values and trigger alarms. Just in case the sensor gives any anomalous values then the values should be eliminated from analysis and triggering of alarms should be stopped. The damaged sensors must get replaced as early as possible. This paper proposes a Data Acquisition System (DAS) which measures humidity, temperature and air quality index at different locations. Just in case the sensor encounters some problem and provides anomalous values, it is detected in real time by comparing these values with predicted values of the 'Moving Average Model' in Python. These anomalous values can then be eliminated from analysis.***
Keywords: Sensors, AQI, Anomalous, DAS, Moving Average Model

I. INTRODUCTION

In various monitoring processes we are able to see that there are various devices which take in and upload the data on the cloud. Anomalies are the deviations of the output values from its normal values. Usually, these devices give outputs in a normal range but because of external factors such as electrical interference, wiring errors we get anomalies within the data. These anomalies can trigger alarms and other processes that are triggered just in case of a hazard. The air quality index depends on many meteorological factors like wind speed, temperature, humidity, rainfall, solar radiation etc. Hence, we cannot keep a fixed threshold for detecting anomalies of these parameters. The threshold needs to be changed according to above mentioned factors. For example, considering only seasonal factors, a threshold value for winters cannot be used for monsoon because some pollutants dissolve in rain- water reducing air quality index. There are many other factors to be considered which makes it difficult to detect anomalies. It is important to detect these anomalies/errors so we do not get any false readings and alarms during the execution. This paper presents a DAS which is developed to measure various parameters like humidity, temperature and air quality at different locations and to detect anomalies in real time.

II. ABBREVIATIONS/ACRONYMS

1) *AQI* – Air Quality Index
2) *CSV* – Comma Separated Values
3) *JSON* – JavaScript Object Notation
4) *ISP* – Internet Service Provider
5) *IP* – Internet Protocol
6) *DAS* – Data Acquisition System
7) *USB* – Universal Serial Bus

III. RELATED WORK

One paper suggested that the petroleum industry is one among the appliance contexts where anomaly problems are present. Luis marti et al. Proposed a mixture of one segmentation algorithm which was a completely unique and fast. They performed a series of empirical studies comparing their approach to other methods applied to benchmark problems and a real-life application related to oil platform turbomachinery anomaly detection. [1]

Another paper suggests that deployment of environmental sensors has generated an interest in real-time applications of the data. David Hill et al. developed a real-time anomaly detection method for environmental data streams that can be used to identify data that deviate from historical patterns. The method was based on an autoregressive data-driven model of the data stream. Furthermore, the proposed method could be easily deployed on a large heterogeneous sensor network. The results indicated that a multilayer perceptron model of the data stream, coupled with replacement of anomalous data points, performed well at identifying erroneous data in the data stream [2].

©IJRASET: All Rights are Reserved

The paper by Octavian A. Postolache presented a network for indoor and out-door air quality monitoring. Each node was installed in a different room and included tin dioxide sensor arrays connected to an acquisition and control system. The nodes were hardwired or wirelessly connected to a central monitoring unit. Advanced processing based on multiple-input–single-output neural networks was implemented at the network sensing nodes to obtain temperature and humidity compensated gas concentration values. [3]. Another paper proposed that to control the energy consumption of a household, sudden abnormal behavior must be detected and adjusted to avoid unnecessary consumption. The thesis presented by Jelena Novacic et al. used a time-series data set of temperature data for implementation of anomaly detection. Four models were implemented and tested; a Linear Regression model, Pandas EWM function, an exponentially weighted moving average (EWMA) model and finally a probabilistic exponentially weighted moving average (PEWMA) model. Each model was tested using data sets from nine different apartments, from the same time period. Then an evaluation of each model was conducted in terms of Precision, Recall and F- measure. The results of this thesis showed that in terms of accuracy, PEWMA outperformed the other models. The EWMA model was slightly better than the Linear Regression model, followed by the Pandas EWM model. [4]

IV. PROPOSED WORK

The aim is to develop a system which can detect anomalies in AQI, humidity and temperature data of different locations in real-time. Detecting anomalies in real-time would prevent triggering of false alarms which occur due to misinterpretation of data. The real-time data is generated by different monitoring devices. Each monitoring device has a micro-controller which gets its current geo-location in terms of latitude and longitude from the nearest cellular Internet Service Provider (ISP) tower and Internet Protocol (IP) address of the network [5]. The geo-location, current I.S.T (Indian Standard Time), and readings of temperature sensor and gas sensor are uploaded onto the Google Firebase account in JavaScript Object Notation (JSON) format [6]. These readings are retrieved using a python module running on a PC and then used for detecting anomalies. Sensor readings are also stored in a Comma Separated Values (CSV) file. The micro-controller uploads data every 20 seconds, hence there are 4,320 readings per day of each parameter. The readings might be missed due to network issues or a power failure. It is important to detect these missed readings and exclude them from the analysis. The time instance at which readings were taken for each device is stored in a hash-map. If the previous time and current time instances match, then it implies few readings are missed. Such readings are excluded from analysis. The predicted values and real-time values for each sensor value are plotted against time on a '*matplotlib*' figure in python [7].

A. Anomaly Detection Algorithm

Moving window averages are used to detect anomalies in the real-time data. The last five readings of each parameter are used to calculate average. For each parameter, a real-time curve of moving averages is plotted. When the absolute difference between the current reading and the parameter's average goes beyond the parameter's threshold, it is detected as an anomaly. The size of the window is five units. Hence for the first five readings,the curves of real- time readings and averages almost coincide with each other. The sensitivity of model to detect anomalies depends on the window size and threshold value. On increasing the window size, the model is unable to detect anomalies properly whereas on decreasing the window size, small changes in parameter are reported as anomalies. While testing the model, it was found that window size of five units was able to detect anomalies precisely for all parameters. Below given is an algorithm for detecting anomalies in one parameter for e.g., humidity. In a similar way, anomalies in other parameters are detected.

1) Declare an empty deque for storing the last five readings of a parameter, arrays for storing real-time readings and average values of the parameter.
2) Declare a hash-map that will keep track of time instance at which last reading was updated by each device.
3) Initialize an endless loop that will keep on running unless the program is terminated.
4) Retrieve the current value of the parameter from google firebase.
5) If the time of current reading is the same as in previous reading of hash-map, print a warning for missed reading and the time instance on console.
6) Else append the parameter's value to the deque as well as to the array storing its real-time values.
7) Calculate the average of the deque. If the absolute difference between the current reading and average is greater than the parameter's threshold, label it as an anomaly. Print the absolute difference, time, and parameter's name on the console.
8) Pop the last value from the deque of that parameter.
9) Write the status to the CSV file.
10) Plot the array of average in red color and parameter's readings against the time axis.
11) At the end of the loop, introduce a delay of 20 seconds which will keep the system synchronized.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

V. IMPLEMENTATION

A. Hardware Requirements

1) NodeMCU ESP8266
2) MQ-135 Gas Sensor
3) DHT-11 Temperature and Humidity Sensor
4) Computing device

B. Software Requirements

1) Python 3.x
2) Arduino IDE

NodeMCU was used as the micro-controller for each device. It has an inbuilt Wi-Fi module. It is cost-effective which is easily programmable via Universal Serial Bus (USB) and requires no external programmer [8]. MQ135 gas sensor gives analog as well as digital readings. For more precision we are using the analog readings. It is sensitive to Benzene, alcohol-based gases, and smoke. It has a fast response time and recovery.

It takes around 20 seconds to calibrate itself [9]. The DHT11 temperature/humidity sensor is a digital sensor. It does not require an external Analog-to-Digital converter.

The temperature range is 0-50ºC which is good enough for monitoring weather. It can measure relative humidity from 20% to 90% [10]. The relative humidity of an air-moisture mixture is ratio of the partial pressure of water vapor in the mixture to the equilibrium vapor pressure of water over a flat surface of pure water at a given temperature. At 100% relative humidity, the air is saturated and is at its dew point.

VI. RESULTS

Each IoT monitoring device had temperature/humidity sensor and gas sensors. The microcontroller in each device was programmed to read the data from the sensors after every twenty seconds. Hence there will be 4,320 total entries for each parameter in 24 hours for each device. Total three such IoT devices were made for generating real-time data. They were connected to the Wi-Fi networks in their locations. The devices uploaded the data on a real-time database of Google Firebase [11]. A maximum delay of three seconds was observed for the devices to upload data once it is collected from sensors.

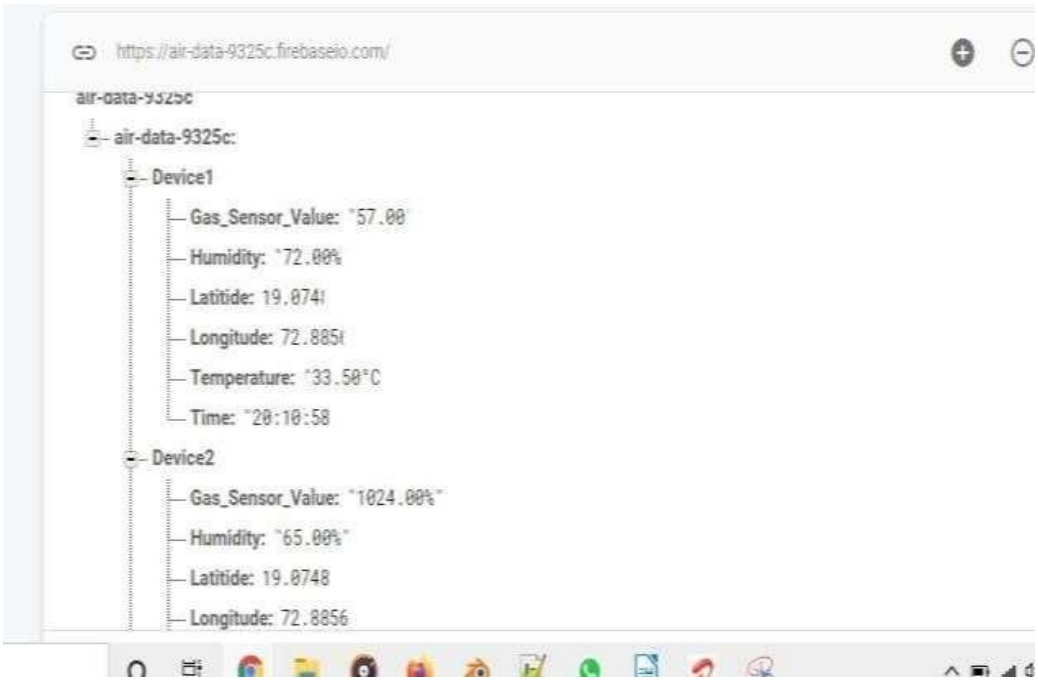


Fig 1: Screenshot of real-time data uploaded on Google Firebase.

The python module retrieves the data after every twenty seconds, plots it, and stores it in a CSV file. Figure 1 above shows the data uploaded by devices on Google Firebase.

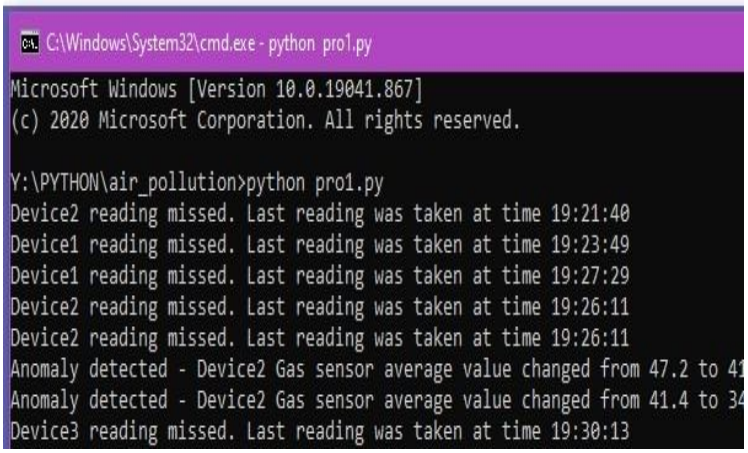


Fig 2: Console status when some readings are missed.

When a reading is missed, or anomaly is detected in parameters it is printed on console. The above Figure 2 shows the console status when some readings are missed or some anomaly is detected. If any anomaly occurs it is classified as temperature anomaly, humidity anomaly or AQI anomaly and recorded separately in another CSV file dedicated to that parameter.

	A	B	C	D	E	F	
1	Device	Time	Air_Quality_Index(ppm)	Temperature(C)	Humidity(%)	AQI_Anomaly	Temp
2	Device1	23:00:08	1024	32.00°C	56.00%		
3	Device2	23:00:11	44	33.20°C	56.00%		
4	Device3	23:00:15	-94	32.50°C	73.00%		
5	Device1	23:00:37	1024	32.00°C	55.00%		
6	Device2	23:00:34	38	33.20°C	56.00%		
7	Device3	23:00:42	-92	32.50°C	73.00%		
8	Device1	23:01:06	1024	32.00°C	56.00%		
9	Device2	23:01:03	44	33.20°C	56.00%		
10	Device3	23:01:04	-97	32.50°C	73.00%		
11	Device1	23:01:30	1024	32.00°C	56.00%		
12	Device2	23:01:33	44	33.20°C	56.00%		
13	Device3	23:01:31	-96	32.60°C	70.00%		
14	Device1	23:01:56	1024	32.00°C	56.00%		
15	Device2	23:01:58	38	33.20°C	57.00%		
16	Device3	23:01:57	-91	32.60°C	73.00%		
17	Device1	23:02:24	1024	32.00°C	56.00%		
18	Device2	23:02:25	44	33.20°C	56.00%		
19	Device3	23:02:23	-94	32.70°C	73.00%		
20	Device1	23:02:49	1024	32.00°C	56.00%		
21	Device2	23:02:59	38	33.20°C	56.00%		
22	Device3	23:02:50	-93	32.80°C	73.00%		
23	Device1	23:03:11	1024	32.00°C	56.00%		
24	Readings missed						

Fig 3: Readings of different devices in a CSV file.

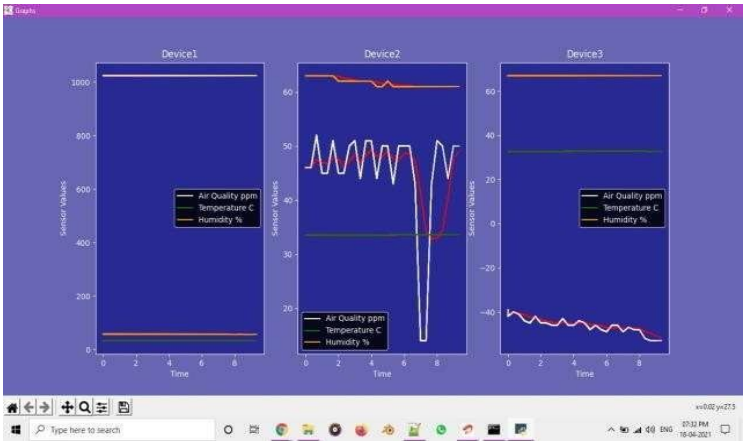


Fig 4: Graphical plot of real-time data from different IoTdevices

The entire system was completed and tested a series of times. The anomalies were intentionally produced in each parameter for testing the model. Different methods were used for producing anomalies in different parameters. An incense stick or a cigarette lighter was suddenly placed near the gas sensor of one of the devices to produce anomaly in gas concentration readings. For testing humidity parameter, the sensor was brought near a vaporizer or a steaming machine for some time. To produce anomaly in temperature data, one of the devices was suddenly kept near a burning object. Figures 3 and 4 show the screenshot of the CSV file and the real-time plots respectively. In the figure 4, the gas sensor of device 2 was temporarily disconnected from the micro-controller to simulate anomaly in Air Quality. As a result, a sudden trough was observed in the air quality. The figures 5-7 show results when vaporizer was placed near the temperature/humidity sensor of device 1 for some time. As a result, the temperature and humidity values shown a sudden spike. Later on, the vaporizer was switched OFF, and the fan in the room was turned ON. The temperature came to a normal value but the humidity fluctuated due to fan. The anomalous values of the sensors in these events were successfully detected by the model. Initially, temperature changed from 34.58ºC to 37.83ºC while humidity changed from 68.2% to 73.8%. Further changes in these values are shown in Figure 6. Figure 5 is the graph of real-time readings. Figures 7a and 7b are temperature and humidity changes recorded in the CSV files dedicated to record only anomalous temperature and humidity values. Figure 8 shows anomalous AQI values recorded in the dedicated CSV file when incense stick was used for simulating anomaly.

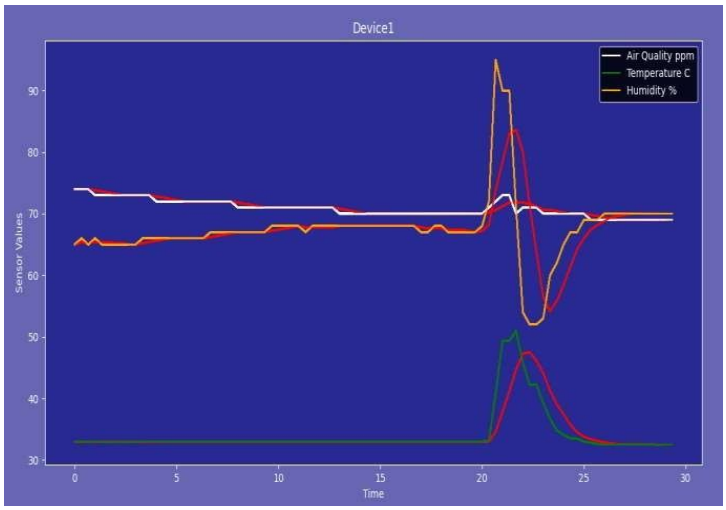


Fig 5: Graph of device 1 readings when vaporizer was usedfor simulating anomaly in humidity and temperature.

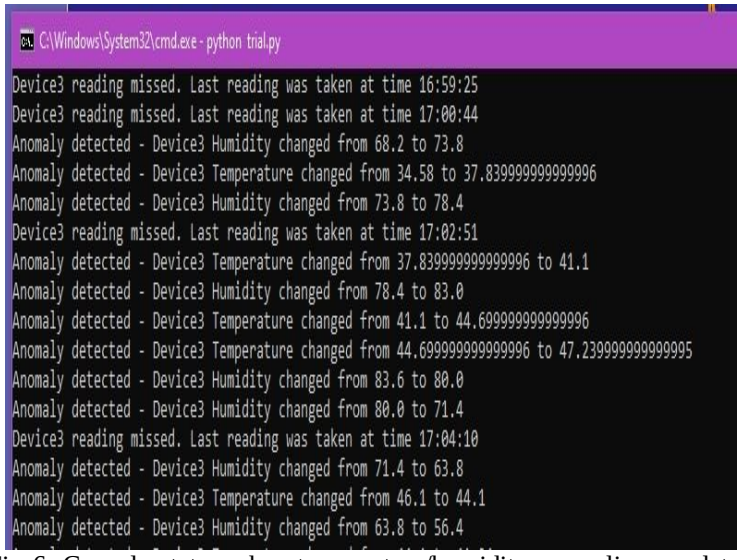
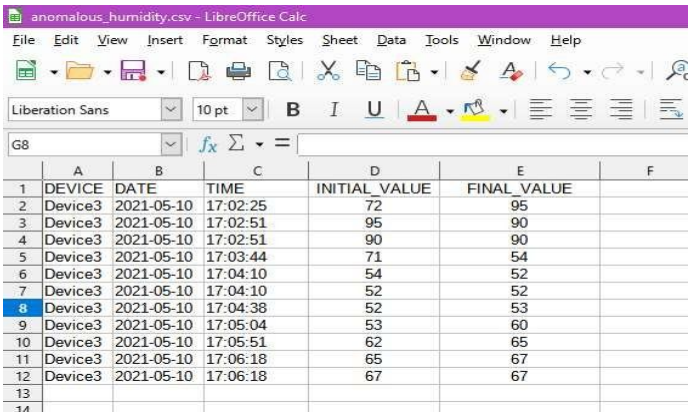


Fig 6: Console status when temperature/humidity anomalies are detected


anomalous_humidity.csv - LibreOffice Calc

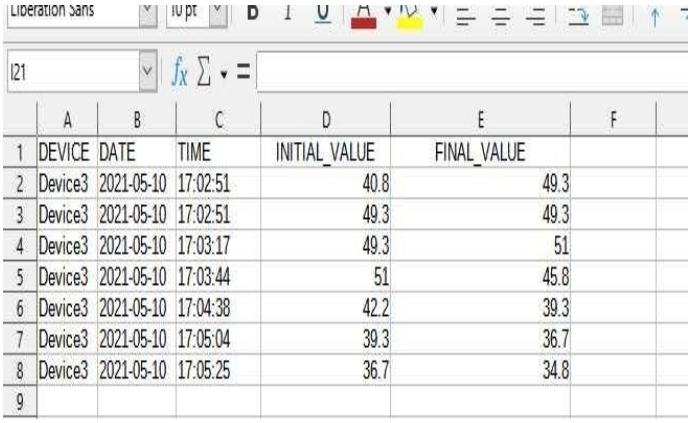
File Edit View Insert Format Styles Sheet Data Tools Window Help

Liberation Sans 10 pt

G8

	A	B	C	D	E	F
1	DEVICE	DATE	TIME	INITIAL_VALUE	FINAL_VALUE	
2	Device3	2021-05-10	17:02:25	72	95	
3	Device3	2021-05-10	17:02:51	95	90	
4	Device3	2021-05-10	17:02:51	90	90	
5	Device3	2021-05-10	17:03:44	71	54	
6	Device3	2021-05-10	17:04:10	54	52	
7	Device3	2021-05-10	17:04:10	52	52	
8	Device3	2021-05-10	17:04:38	52	53	
9	Device3	2021-05-10	17:05:04	53	60	
10	Device3	2021-05-10	17:05:51	62	65	
11	Device3	2021-05-10	17:06:18	65	67	
12	Device3	2021-05-10	17:06:18	67	67	
13						

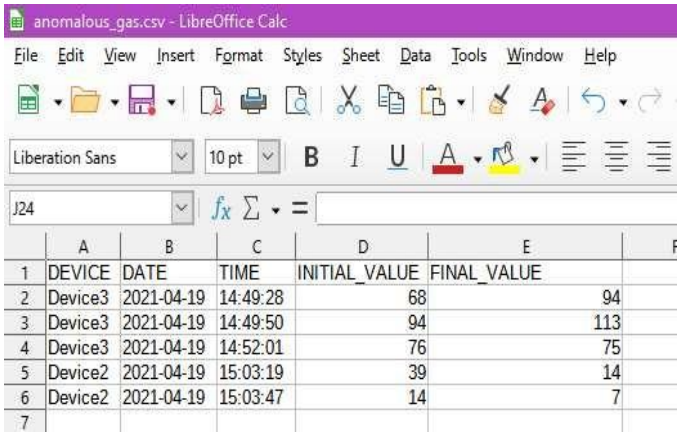
Figure 7a: Recordings of anomalous humidity values in itsdedicated CSV file.


Liberation Sans 10 pt

I21

	A	B	C	D	E	F
1	DEVICE	DATE	TIME	INITIAL_VALUE	FINAL_VALUE	
2	Device3	2021-05-10	17:02:51	40.8	49.3	
3	Device3	2021-05-10	17:02:51	49.3	49.3	
4	Device3	2021-05-10	17:03:17	49.3	51	
5	Device3	2021-05-10	17:03:44	51	45.8	
6	Device3	2021-05-10	17:04:38	42.2	39.3	
7	Device3	2021-05-10	17:05:04	39.3	36.7	
8	Device3	2021-05-10	17:05:25	36.7	34.8	
9						

Fig 7b: Recordings of anomalous temperature values in its dedicated CSV file.


anomalous_gas.csv - LibreOffice Calc

File Edit View Insert Format Styles Sheet Data Tools Window Help

Liberation Sans 10 pt

J24

	A	B	C	D	E	F
1	DEVICE	DATE	TIME	INITIAL_VALUE	FINAL_VALUE	
2	Device3	2021-04-19	14:49:28	68	94	
3	Device3	2021-04-19	14:49:50	94	113	
4	Device3	2021-04-19	14:52:01	76	75	
5	Device2	2021-04-19	15:03:19	39	14	
6	Device2	2021-04-19	15:03:47	14	7	
7						

Fig 8: Recordings of anomalous AQI values in its dedicated CSV file.

VII. CONCLUSIONS

The anomaly detection system is successfully able to detect anomalies in the readings of gas sensors, temperature and humidity sensors in real-time. The system plots a real-time graph of all the readings which changes every twenty seconds. The working of the entire system is tested by intentionally creating the conditions in which anomalies are likely to occur. Anomalies are successfully detected and can be removed from analysis in real-time. This would prevent the triggering of false alarms in a factory or any other air pollution monitoring system. Filtering of data is essential before the analysis of any sensor data, but the data generated by the anomaly detection system can be directly used for analysis without filtering. The concept of detecting anomalies in real-time can be further extended to other real-time monitoring systems as well.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

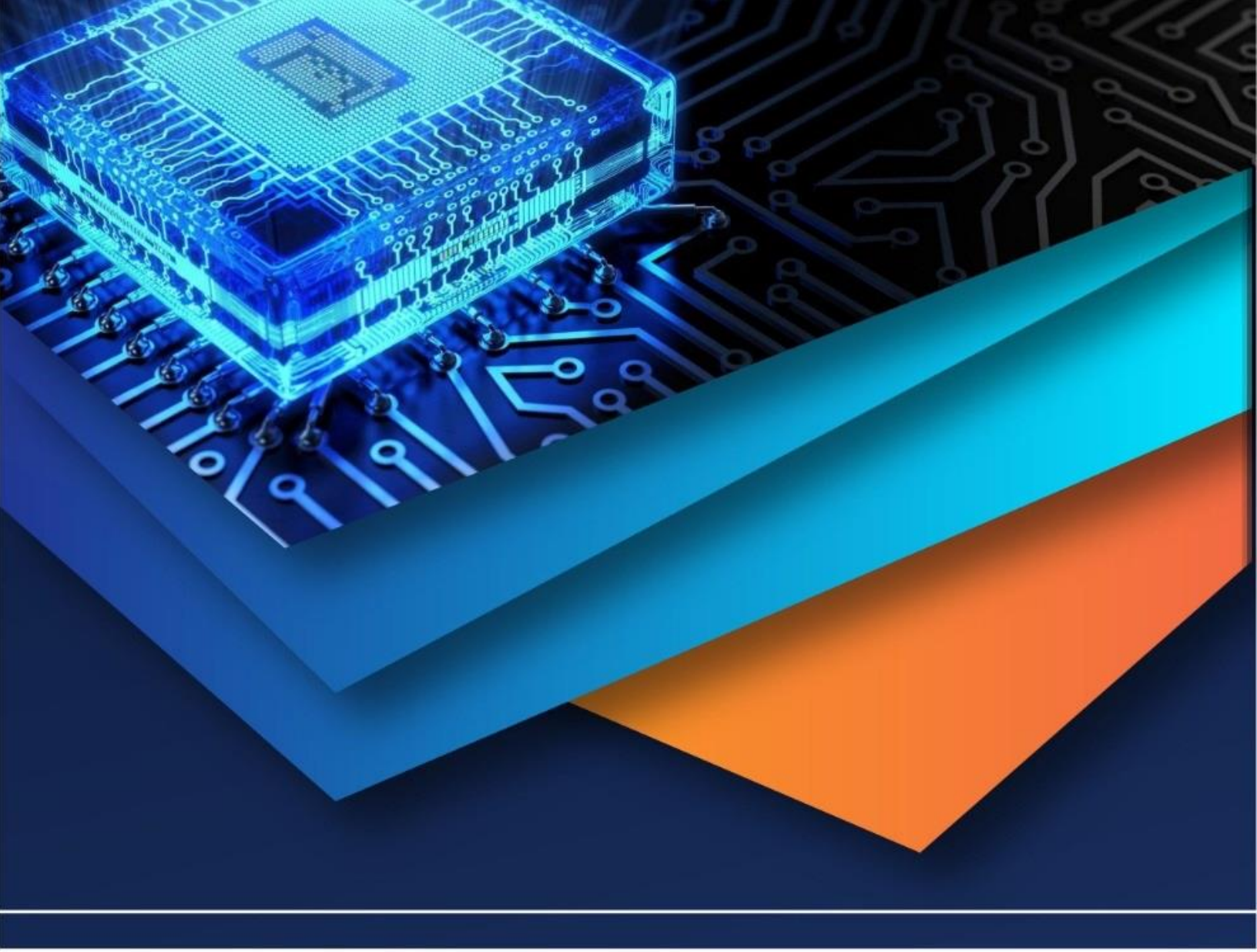
VIII. ACKNOWLEDGEMENT

The support provided by Electronics and Computer Science Engineering Department of Fr. Conceicao Rodrigues College of Engineering, University of Mumbai is greatly appreciated.

REFERENCES

[1] L. Martí, N. Sanchez-Pi, J. Molina, and A. Garcia, "Anomaly Detection Based on Sensor Data in Petroleum Industry Applications," Sensors, vol. 15, no. 2, pp. 2774–2797, Jan. 2015.
[2] David J. Hill, Barbara S. Minsker, "Anomaly detection in streaming environmental sensor data: A data-driven modeling approach", Environmental Modelling& Software, Volume 25, Issue 9, 2010, Pages 1014-1022, ISSN 1364-8152, https://doi.org/10.1016/j.envsoft.2009.08.010. (https://www.sciencedirect.com/science/article/pii/S1364815 209002321)
[3] O. A. Postolache, J. M. Dias Pereira and P. M. B. Silva Girao, "Smart Sensors Network for Air Quality Monitoring Applications," in IEEE Transactions on Instrumentation and Measurement, vol. 58, no. 9, pp. 3253-3262, Sept. 2009, doi: 10.1109/TIM.2009.2022372.
[4] Jelena Novacic Kablai Tokhi, "Implementation of Anomaly Detection on a Time- series Temperature Data set", B.S Thesis, Department of Computer Science and Media Technology, Malmo University, Malmö, Sweden, 2019. [Online]. Available: http://ls00012.mah.se/handle/2043/28784
[5] IP API, IP Geolocation API. Accessed – January 12, 2021.[Online]. Available: https://ip-api.com/docs
[6] Rui Santos and Sara Santos. "Get Epoch/Unix Time withthe ESP8266 NodeMCU Arduino" randomnerdtutorials.comhttps://randomnerdtutorials.com/epoch-unix-time-esp8266- nodemcu-arduino (accessed Feb. 25, 2021).
[7] Sentdex, United States. Matplotlib Tutorial 16 - Live Graphs(July 12, 2015). Accessed: March 7, 2021. [Online Video].Available: https://www.youtube.com/watch?v=ZmYPzESC5YY
[8] Espressif Systems, "ESP8266EX Datasheet", Dec 2015[Revised Feb. 2018.]
[9] HANWEI ELECTRONICS LTD, "MQ135 Gas SensorTechnical Data", May 2013
[10] Mouser Electronics, "DHT11 Humidity & TemperatureSensor", October, 2018.
[11] Google Firebase, Google. Accessed: Sept. 14, 2020.[Online]. Available: https://firebase.google.com/docs

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)