Air Pollution Detection System using Edge Computing

Katalina Biondi[1], Eyhab Al-Masri[1], Orlando Baiocchi[1], Suganya Jeyaraman[1], Eric Pospisil[1], Graham Boyer[1], Cleonilson Protasio de Souza[2]
[1] School of Engineering and Technology, University of Washington Tacoma, Tacoma, WA USA
{kbiondi, ealmasri, baiocchi, suganjoe, epos33, gboyer01}@uw.edu

[2] Department of Electrical Engineering Federal University of Paraíba, João Pessoa, PB Brazil {protasio@cear.ufpb.br}

***Abstract*—Existing solutions to measuring air quality can be expensive and potentially mutes high air pollution events. The IoT Pollution Project is exploring how IoT concepts can be applied with smart systems to detect pollution in real-time. Using a network of Raspberry Pi prototypes, the project aims to measure heavily populated areas around the City of Tacoma, while building a real-time interface measuring current air quality. The project also explores the use of edge computing as an alternative to cloud computing. The vast expansion of IoT devices poses threats to the infrastructure of cloud computing as more devices process and store data to the cloud. The project demonstrates how edge devices can alleviate the work done on the cloud by calculating rolling averages over a time interval on the edge device and then deploying the data to the cloud. The project uses Microsoft Azure Framework, IoT concepts and edge computing concepts to build the project architecture.**

Keywords— Internet of Things, Raspberry Pi, Microsoft Azure, LoRa, Pollution Detection, Environmental Monitoring, Real-Time Data, IoT, Edge Computing, Cloud Computing, Air Quality

I. INTRODUCTION

Air quality and pollution monitoring have become significant over the years due to the increase in premature mortality rate. Historically, some of the main causes of poor air quality are rapidly developing industry fuels as a result of high levels of smoke and sulfur dioxide through combustion [13]. Traffic emissions also emit levels of various pollutants such as carbon monoxide and particulate matter PM10. Forest fires have also recently played a role in contributing to high levels of pollutants including Particulate Matter 2.50 which causes many health issues. Health issues resulting from moderate to high levels of poor air quality can result in complications in respiratory and inflammatory response systems, more seriously can lead to heart disease and cancer [20].

In addition, premature deaths are rapidly occurring around the world due to the increase in poor air quality. According to the World Health Organization, about seven million deaths occur around the world every year as a result of exposure to pollution, some of which are from ambient (outdoor) air pollution and other deaths are the result of household exposure to smoke from dirty cook stoves and fuels [20]. One of the major challenges in detecting air pollution is that it varies from location to another. That is, pollution, for example, may buildup in pockets within a centralized area and large pollution-related events often result in pollution traveling over geographic distances. Hence, this pollution-related variation often leads to uncertainty in the cause of pollution as well as unawareness that the current air quality levels have changed. To this extent, measuring air quality index while continuously monitoring pollution becomes critical for detecting environmental changes over time.

There are two main methods for measuring air pollution: (a) through satellite-based imagery and (b) through ground-based measures. The satellite-based imagery is more complex due to the fact it requires deployment of satellites, special equipment for using proper communication channels and powerful hardware that is required for processing images. The ground-based measurement systems may be less costly compared to that of the satellite-based imagery. However, the deployment of pollution sensors across a large geographic area can be quite expensive. In addition, the two methods vary in terms of the frequency of collecting air pollution measurements. For example, satellite-based methods can provide measurements for particle pollution every few minutes (e.g. five minutes) throughout the day.

Furthermore, other pollution methods focused on processing historical air pollution data and use time-series analysis (e.g. averages from previous years and previous time segments) [15]. Although this is an efficient method of measuring pollution averages, this method does not include the context in which pollution has occurred (e.g. exclude the causes that lead to pollution) and adds some ambiguity in pollution risk locations as to where and when the pollution has occurred. Implementing a real-time pollution detection system would allow smart cities, for example, to monitor pollution levels in real-time which can enable the identification of contextual information such as (a) location, b) time and (c) potential cause of large pollution events. Not only is identifying the time and location of substantial pollution events beneficial, but it also contributes to the planning of reduce short- and long-term health effects during which these pollution events occur.

To overcome many of the existing challenges associated with detecting air pollution, in this paper we introduce a real-time Internet of Things (IoT)-based system that provides a cost-effective solution for monitoring and identifying major air pollution events. One of the main goals of this system is to explore alternative solutions to emerging problems with by employing Internet of Things (IoT) and cloud computing capabilities.

Cloud computing provides a cost-effective way for storing and processing large volumes of data [2]. In recent years, cloud computing has become an effective tool for employing IoT devices to sense and collect data over large geographic distances. However, utilizing cloud-based technologies to an air pollution system may not always be ideal. As the number of IoT devices increase, the ability control and maintain large number of IoT devices becomes a challenging problem. In addition, Cisco IBSG predicts there will be 50 billion devices connected to the Internet by 2020 [4] which will introduce latency issues related to the deployment of IoT devices.

The number of devices accessing the cloud at this large scale poses many new challenges and risks for the cloud infrastructure. Edge computing provides a solution to some

of these challenges. Through the utilization of edge computing concepts with the deployment of these IoT devices, edge computing theoretically will supplement some of the processing of data originally performed on the cloud. For example, data will be processed and collected at the edge of the network. This means that IoT devices can process data locally using their own processing power and then deploy a subset of desired data to the cloud. Hence, rather than deploying all raw data, only a subset of the data is sent to the cloud. Processing data at the network edge would yield shorter response times, reduces network latency and increases client [2]. Although edge computing offers some advantages to the challenges in detecting air pollution in real-time, there are still some challenges posed by this new technology.

Edge computing poses new challenges for IoT systems. Drawbacks include, for example, limited computing resources of some edge computing devices. That is, since processing may take place at the edge device or node, this translates into computational processing power that is required at the edge device. Furthermore, the size of these devices will often increase and the power required to run these devices will eventually lead to more power required for processing. [18].

Throughout this paper, we present an air pollution detection system that aims to continuously monitor air quality in real-time. In addition, the system will keep track of the location at which these air quality measurements occur. The system uses low-cost hardware prototyping platforms as a proof of concept and integrates the use of edge computing concepts. The system is comprised of Raspberry Pi which acts as an edge computing device that communicates with the Microsoft Azure platform cloud-based computing platforms. The system has been tested through its deployment at the University of Washington Tacoma campus in order to collect and analyze data in a real-time manner. In the future deployments of this system, we plan to integrate long-range wireless communication protocols such as LoRa. Using LoRa it is then possible to transmit data over long ranges or distances while minimizing the power consumption for transmitting data over long distances [17]. Hence, this means that air pollution detection system could cover a larger range of the City of Tacoma and the battery life of the devices would be more efficient.

This paper is organized as follows; Section two discusses some of the existing solutions or related work. Section three introduces the hardware and the system architecture. Section four provides a discussion of our proposed system and finally, Section five provides a conclusion and future work.

II. Related Work

There are many alternatives to the hardware and software architecture for this project. The project incorporates a Raspberry Pi; however, there are other hardware options to consider. Similar projects have used other hardware, such as Arduino, however most research has suggested that the Raspberry Pi is a reliable choice. Shete R. and Agrawal S. conducted a similar framework that provides monitoring pollution in city environments. They concluded that Raspberry Pi was among the better hardware because it allows for low cost and high latency. The Raspberry Pi can act as an effective server and provides reliable internet connectivity, whereas other micro-controllers may need an extra Ethernet module to get internet connectivity [14].

Other research efforts have implemented different types of computation for data collection and processing. As mentioned earlier, it is common to utilize cloud computing with IoT devices. The authors in [8] implemented a similar framework with cloud computing using Raspberry Pi and Arduino and IBM Bluemix IoT platform [8]. Cloud computing can be used for air pollution detection system to handle the storage, management and processing of data with minimal effort [10]. However, with the increasing volume of data collection in a real-time manner that may be used for further processing requires more intelligent devices that do not solely depend on the cloud for managing and processing data. Cloud computing makes data collection and processing manageable, however, the trade-off as discussed earlier is the risk of having increased network latency due to large number of devices accessing the network.

To account for the rapid growth of IoT devices, research involving edge computing has become an increasingly important alternative to cloud computing. Edge computing refers to the enabling technologies allowing computation to be performed at the edge of the network [6, 22-25]. In [6], the authors developed a prototype for the evaluation of pollution using an Arduino board and IBM Watson IoT platform. Using edge computing, their model is structured to reduce the computational burden on the cloud [6].

Although edge computing can be beneficial for the deployment of some IoT-application, there are also challenges associated with edge computing. Some of these challenges include the risk for data leaks and security breaches. Furthermore, edge devices require power and internal processing compared to smaller IoT devices that send data directly to the cloud without internal processing. Therefore, both edge and cloud computing technologies have unique challenges associated with the deployment of IoT-based applications. To this extent, in this paper we present an exploratory study using a proof of concept for the deployment of an air pollution system that utilizes edge computing capabilities for environmental air pollution monitory.

III. Prototype Architecture

A. Hardware Architecture

The prototype hardware consists of a Raspberry Pi 3 paired with a Grove Pi, Grove Dust sensor and a Grove Air Quality Sensor. The Grove Pi is a modular system that allows sensors to be connected to the Raspberry Pi board while minimizing wire connections [16]. The Grove Pi is mounted on top of the Raspberry Pi 3 through the GPIO pins. The Grove dust sensor and the Grove air quality sensor is attached to the Grove Pi as shown in Fig. 1.

The Grove dust sensors are used to measure the dust concentration using particulate matter levels in the air. These sensors are used to calculate the concentration by using the low pulse occupancy time in a given unit. This low occupancy time is proportional to PM concentration. The sensors are most responsive to PM2.5 fine particles with a diameter $\leq 2.5\mu m$ [21]. The Grove air quality sensor monitors the air condition by measuring a comprehensive scope of harmful gases such as carbon monoxide [16]. This air quality sensor is a general-purpose sensor that does not

provide specific data that pertain to the specific gas concentration. However, it is used to provide a qualitative measure indicating whether the air measurements are clean or polluted depending on the levels reported by the sensor [3]. The air quality sensor on the Grove air quality board v1.3 contains an air-quality gas sensor, which reports a voltage level that can be used as an indication of the air quality level. We used the quantitative interpretation of the values reported by the Grove air quality sensor v1.3 based on the metrics reported in [27] and are shown in Table 1.

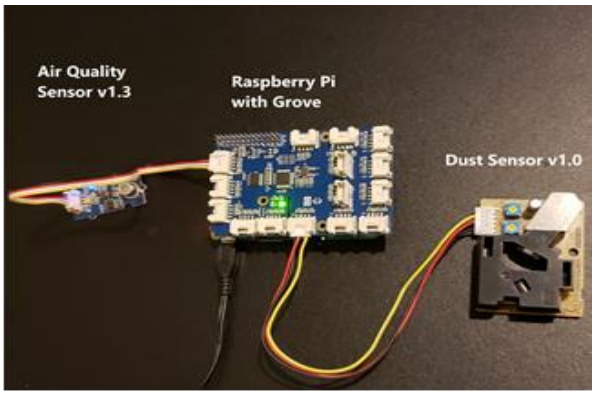

Figure 1: *Air Pollution* Hardware *Prototype using Raspberry Pi and GrovePi Modules*

B. *System Architecture*

The IoT Pollution Detection Project implements Microsoft Azure framework to control a system of Raspberry Pi devices from one centralized environment. Fig. 2 presents the application layer for our air pollution detection system. Through this layer, it is then possible to connect multiple edge devices (e.g. Raspberry Pi) with a cloud-computing platform (e.g. Microsoft Azure framework). The devices are all configured and communicate with the Azure framework in real-time. Using Microsoft Azure cloud and the Microsoft Azure IoT Hub, all the edge devices can be controlled and monitored from one platform.

To secure a connection between the Azure IoT hub and the Raspberry Pi edge deceives, a primary key/connection string is added into the configuration file of the Raspberry Pi devices when created in the hub. The system architecture then utilizes the Azure Container Registry to store docker container modules. Using Visual Studio Code IDE, the necessary scripts and modules are developed and deployed to all the edge devices.

The system also uses the docker modules Air Quality and Geo-location. Through these modules, the system collects data by reading the air quality and taking a rolling average after five seconds. The primary data of interest is when levels being measured reach above a threshold amount. When this threshold is breached, the data collected is then sent to the cloud. The Geo-location Module contacts the Bing Maps API and gets the current location of the Raspberry Pi device. This location is essential to the design because the system can identify where the air pollution events are at that current time. After the modules are developed, they are packaged and sent to the Microsoft Azure Container Registry. To make the deployment process more efficient across a system of multiple edge devices, a Deployment Manifest File is created. This file is a JSON document that describes IoT Edge agent and the IoT Edge Hub module twin. This document includes credentials to access private containers as well as how to create and manage each of the modules created. It also manages the flow between these docker modules and the IoT Hub [7].

The architecture utilizes multiple modules that work together to collect data, the deployment manifest file is used to deploy all necessary configurations and instructions to all the edge devices in the system at once. Now the Raspberry Pi system identifies where to pull the docker modules, it can simulate the instructions on each individual Raspberry Pi. The modules instruct the edge devices to send an air quality message to the cloud if any of the Raspberry Pi's in the system measure a reading above the set threshold.

Range	Quantitative Interpretation
0-3199	Fresh Air
3200 – 4799	Low Pollution
4800-6399	Medium Pollution
6400-11199	High Pollution
111200 and more	Very High Pollution

Table 1: Air Quality Range Interpretation based on Metrics Reported in [27].

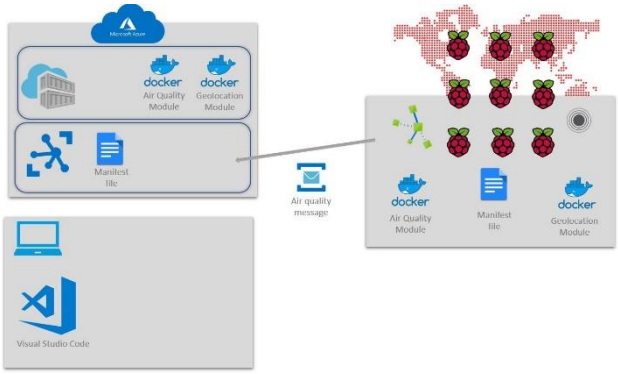

Figure 2: Application Layer for Air Pollution Detection System

Once the IoT Hub receives data from the edge devices, the architecture begins the post-processing of the data. Fig. 3 shows the interactions between different platforms during the processing of the data. The architecture utilizes a message queuing system and data transformation tools provided by Microsoft Azure. Microsoft Azure provides a Message Bus Queue that receives messages and processes them in a first in first out basis. When the hub receives a message from the Raspberry Pi, those messages are sent to and received from queues.

The Queues enable the architecture to store the messages until the Stream Analytics and the built-in Logic App is available to receive and process them [19]. Using the Microsoft Azure Logic App, a generated message was built that will send an alert message to a specified email if the data ever breach the threshold. Fig. 4 Message Alert System demonstrates the process of setting up these alerts. The data sent to the hub is analyzed with Microsoft Stream Analytics, which simply integrates the IoT Hub and provides real-time analytics on the data received [1]. All the data is stored long term, therefore large storage is necessary for this application and Microsoft Azure SQL DB storage is used in the architecture. Finally, the architecture uses Stream Analytics to create a Power BI dashboard, which gives live command and control view and that shows the real-time data from the

system while also using maps to show the current location of each Raspberry Pi.

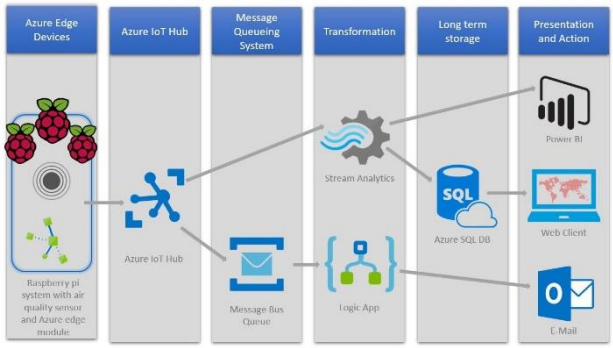

Figure 3: High-Level Overview of Data Flow

IV. Discussion

The system successfully configures a Raspberry Pi system containing multiple edge devices as a proof of concept. From the Microsoft Azure IoT Hub portal, details about the edge devices can be viewed. Using the Device Details portal in Azure, it was verified that the Primary Key and the Connection String that is required for the communication between the edge device and the IoT hub were successfully deployed to each Raspberry Pi in the system. The system verifies that the modules were successfully running on the system by using the Linux command, sudo IoT edge list. After running the Python Module, successful messages were generated from the data collected from the device. Fig. 5 Python Module Output shows the output of these messages. They include the location of the device, when the message was created, the message Id, which device generated the message, and the metrics for the dust and air quality.

On the IoT Hub, there is a graphical representation on how many telemetry metrics were sent to the hub from the edge devices over an interval of time. Fig. 7 Telemetry Messages shows an example of the total amount of telemetry messages sent over an interval of thirty seconds. The graph shows the time when messages were sent from the device, and how many messages were sent. This interval produced a sum of 695 telemetry message sent to the IoT Hub. This graph is just another demonstration that acknowledges the successful communication through the edge deceives and the IoT hub. Another acknowledgment that the architecture was working successfully was through Visual Studio Code IDE.

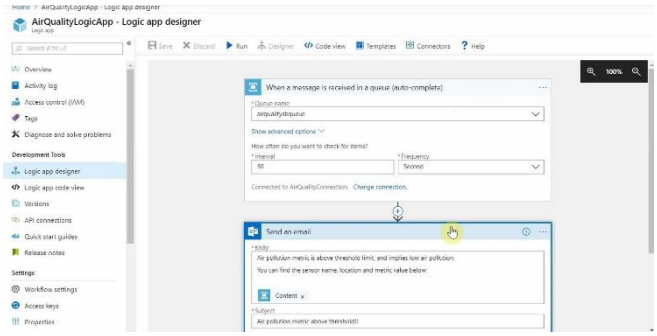

Figure 4: Message Alert System

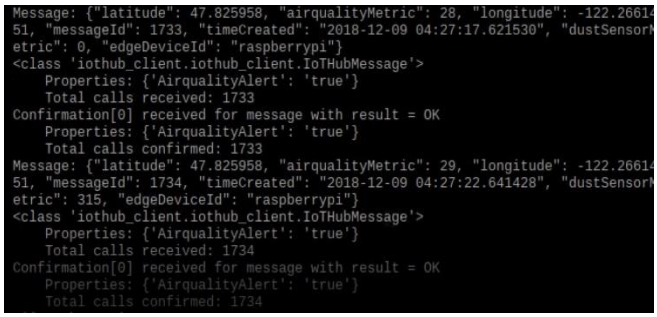

Figure 5: Snippet of the Python Module Output

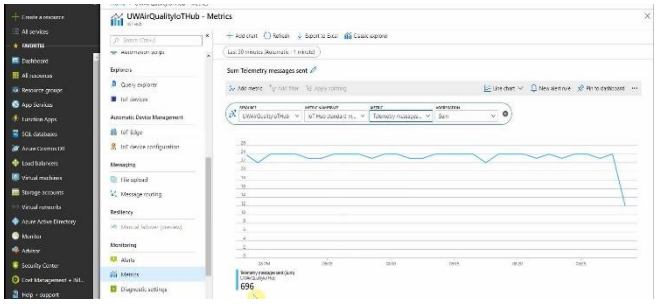

Figure 6: Dashboard Module Displaying Telemetry Messages

Logging into Visual Studios using the same email used for the Microsoft Azure account allows visual access to the Azure IoT Hub devices. From this platform, it was verified that all the modules are currently running. Visual Studios has a tool Start Monitoring D2C Messages, through this tool users can monitor the device to cloud messages in the output console box.

Fig. 7 shows a snippet of all messages received from the Raspberry Pi system in real- time. Every time a new message is generated, the platform displays the new message. The project also successfully managed data visualization through the Power Bi Dashboard. On the Power Bi dashboard, four charts are initially displayed. Each chart represents either the average measurements of air quality, the average measurement of dust quality (PM), and the location of each sensor.

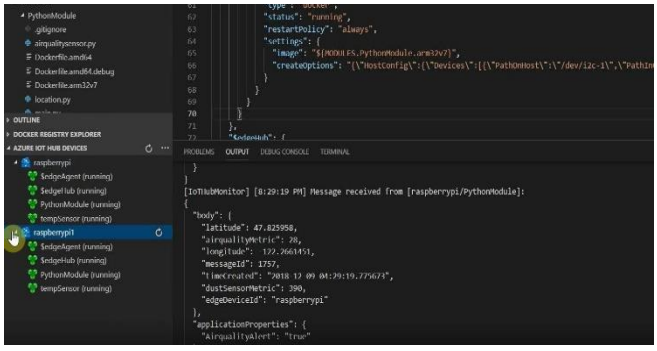

Figure 7.Visual Studio Console Output

Fig. 8 presents the Power BI dashboard for our IoT edge-based system. The graphs assign different colors to the different edge devices, so it is easier to distinguish which device recorded the real-time data. Below each average metric, the user can visually see the location of where the average readings are coming from in real-time.

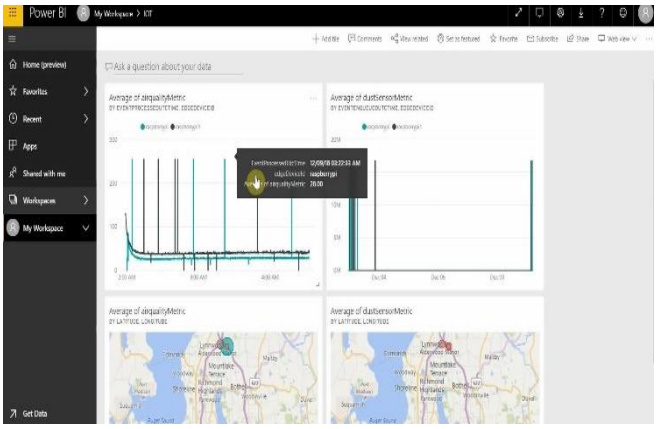

Figure 8: Air Pollution IoT Edge-Based System Power BI Dashboard

V. Conclusion

Throughout this paper, we presented a system that demonstrates usefulness of utilizing edge computing for building an air pollution detection system. We use this system as a proof of concept to demonstrate that multiple edge devices equipped with air pollution sensors can be used effectively to monitor the air quality levels covering a small geographic area or distance in a real-time manner. Throughout the deployment of our system, we focused on the integration of edge computing capabilities combined with cloud computing capabilities. In addition, we focused on building a cost-effective solution that utilizes low-cost edge devise for the purpose of monitoring air pollution in real-time.

The results provided in this paper demonstrate the effectiveness of using an edge-based solution for the purpose of monitoring air pollution levels. Furthermore, it also demonstrates that creating a network of low-cost edge devise with sensors can be used to obtain air pollution data in a real-time manner. This data can then be further used for additional insights or decision-making. In addition, integrating the location at which this data is generated is vital to determine any pattern in air pollution changes. That is, the contextual information collected provides more insights about the building of large pollution events. This information can be useful to applications such as smart cities.

In this paper, we also presented the advantages and disadvantages by integrating cloud computing and edge computing capabilities for the purpose of monitoring air pollution in real-time. The fact that the system is cost-effective translates into ease of deployment and being accessible to many users. However, utilizing edge computing imposes some restrictions such as the increased size of the IoT devices, which require power processing, power. Furthermore, another limitation that was discussed throughout the paper is the fact that utilizing communication protocols may have some restriction on how far data can be transmitted. To this extent, we aim as a future work to extent this system to support long-range transmission using protocols such as LoRa.

Acknowledgment

This project is partially supported by the Office of Research – Collaborative Publicly Engaged Scholarship (CPES) (2018) and the Student Technology Fee (STF) funds at the University of Washington Tacoma (19A018).

References

[1] Azure.microsoft.com, 'Stream Analytics - Real-time data analytics — Microsoft Azure' [Online]. Available: https://azure.microsoft.com/en-us/services/stream-analytics/ [Accessed: 04- 05- 2019].

[2] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. 2012. Fog computing and its role in the internet of things. In Proceedings of the first edition of the MCC workshop on Mobile cloud computing (MCC '12). ACM, New York, NY, USA, 13-16.

[3] Grove Air Quality http://wiki.seeedstudio.com/Grove-Air_Quality_Sensor_v1.3/ [Accessed: 04- 05- 2019].

[4] Evans, D. (2011). The internet of things: How the next evolution of the internet is changing everything. *CISCO white paper*, *1*(2011), 1-11.

[5] Gartner Inc., The Impact of the Internet of Things on Data Centers, Retrieved from https://www.gartner.com/en/documents/2672920[Accessed: 04- 05- 2019].

[6] Idrees, Z., Zou, Z., & Zheng, L., Edge Computing Based IoT Architecture for Low Cost Air Pollution Monitoring Systems: A Comprehensive System Analysis, Design Considerations & Development. *Sensors*, *18*(9), 3021, 2018.

[7] Kgremban, 'Declare Modules and Routes with Deployment Manifests - Azure IoT Edge. Declare Modules and Routes with Deployment Manifests - Azure IoT Edge — Microsoft Docs' [Online]. Available: https://docs.microsoft.com/en-us/azure/iot-edge/module-composition. [Accessed: 28- Feb- 2019].

[8] Kumar, S., & Jasuja, A. (2017, May). Air quality monitoring system based on IoT using Raspberry Pi. In 2017 International Conference on Computing, Communication and Automation (ICCCA) (pp. 1341-1346). IEEE.

[9] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," IEEE Pervasive Comput., vol. 8, no. 4, pp. 14–23, Oct./Dec. 2009.

[10] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," https://www.nist.gov/sites/default/files/documents/itl/cloud/cloud-def-v15.pdf [Accessed: 04-05-2019]

[11] Mikejo5000, 'ClickOnce Deployment Manifest - Visual Studio' [Online]. Available : https://docs.microsoft.com/en-us/visualstudio/deployment/clickonce-deployment-manifest?view=vs-2017 [Accessed: 28- Feb- 2019].

[12] Morabito, Roberto & Beijar, Nicklas. (2016). Enabling Data Processing at the Network Edge through Lightweight Virtualization Technologies. 10.1109/SECONW.2016.7746807.

[13] Oxfordshire.air-quality.info, 'Why Air Quality is important' [Online]. Available: https://oxfordshire.air-quality.info/why-air-quality-is-important [Accessed: 04- 05- 2019].

[14] R. Shete and S. Agrawal, "IoT based urban climate monitoring using Raspberry Pi," 2016 International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur, 2016, pp. 2008-2012.

[15] Scijinks.gov, 'How Is Air Quality Measured?' [Online] Available: https://scijinks.gov/air-quality/ [Accessed: 04- 05- 2019].

[16] Seeedstudio.com, 'GrovePi' [Online]. Available: https://www.seeedstudio.com/GrovePi-p-1672.html [Accessed: 04-05-2019].

[17] Semtech.com, 'What Is LoRa?' [Online]. Available: www.semtech.com/lora/what-is-lora [Accessed: 28- Feb- 2019].

[18] Song, Y., Yau, S-S., Yu, R., Zhang, X., & Xue, G. (2017). An Approach to QoS-based Task Distribution in Edge Computing Networks for IoT Applications. In *Proceedings - 2017 IEEE 1st International Conference on Edge Computing, EDGE 2017* (pp. 32-39). [8029254] Institute of Electrical and Electronics Engineers Inc.. https://doi.org/10.1109/IEEE.EDGE.2017.50

[19] Spelluru, 'Azure Service Bus Messaging Documenta-tion - Tutorials, quickstarts, API Reference' [Online]. Available: https://docs.microsoft.com/en-us/azure/service-bus-messaging [Accessed: 28- Feb- 2019].

[20] Who.int, 'Air Pollution', 2019. [Online]. Available: https://www.who.int/airpollution/en/ [Accessed: 04- 05- 2019].

[21] wiki.seeedstudio.com, 'Grove - Dust Sensor' [Online]. Available: http://wiki.seeedstudio.com/Ghttp://wiki.seeedstudio.com/Grove-Dust_Sensor/rove-Dust_Sensor/[Accessed: 04- 05- 2019].

[22] M. Satyanarayanan, “The emergence of edge computing,” Computer, vol. 50, no. 1, pp. 30 – 39, Jan 2017.

[23] P. Garcia Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, and E. Riviere, “Edge-centric computing: Vision and challenges,” SIGCOMM Comput. Commun. Rev., vol. 45, no. 5, pp. 37 – 42, Sep. 2015.

[24] Q. Pu, G. Ananthanarayanan, P. Bodik, S. Kandula, A. Akella, P. Bahl, and I. Stoica, “Low latency geo-distributed data analytics,” SIGCOMM Comput. Commun. Rev., vol. 45, no. 4, pp. 421 – 434, Aug. 2015.

[25] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, \Edge computing: Vision and challenges,” IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637 – 646, Oct 2016.

[26] Air-Quality Gas Sensor, Zhengzhou Winsen Electronics Technology Co, https://www.winsen-sensor.com/d/files/PDF/Flat-surfaced%20Gas%20Sensor/MP503%20Manual%20(Ver1.4).pdf, [Accessed: 04- 05- 2019].

[27] Grove Air Quality Sensor, Air Quality Sensor Report https://shop.switchdoc.com/products/grove-air-quality-sensor-1, [Accessed: 04- 05- 2019].