On Construction of Sensors, Edge, and Cloud (iSEC) Framework for Smart System Integration and Applications

Endah Kristiani, *Graduate Student Member, IEEE*, Chao-Tung Yang, *Member, IEEE*, Chin-Yin Huang, Po-Cheng Ko, and Halim Fathoni

***Abstract*—Intelligent systems influence many aspects of daily life. With the emergence of the Internet of Things (IoT), artificial intelligence (AI), and machine learning (ML), opportunities have been created for smart computing infrastructure. However, problems might arise from the lack of interconnectivity, higher reliability, real-time predictive analytics, and low-latency requirements. Therefore, in this article, we propose the sensors, edge, and cloud (iSEC) framework. The project deploys a smart cloud edge-computing architecture to provide ML and deep learning in the cloud edge environment. Two pilot projects of air quality monitoring system and object detection are demonstrated to evaluate the iSEC framework.**

***Index Terms*—Cloud computing, edge computing, Internet of Things (IoT), long-range low-power wide-area network (LPWAN), message passing interface (MPI).**

I. Introduction

THE INTEGRATION of cloud computing, edge computing, and Internet-of-Things (IoT) technology establish a smart cloud edge-computing environment. Combine with the design of a machine learning (ML) and deep learning framework to deploy theoretical applications in the real world [1]–[3]. In this case, the success of the IoT applications depends on the ability to monitor, manage, and verify [4] the system. The implementation of the IoT often has to be supported and maintained through a large number of sensors and big data analytics tools. It is essential to design how to perform further real-time analyses, such as anomaly detection, object identification, and emergency situation [5], [6]. The emergence of smart IoT devices has led to the definition of the Medical IoT (IoMT), which is revolutionizing the way healthcare is tackled worldwide. The data generated by wearable devices is growing and can be spread between clinical centers, hospitals, and research laboratories [7], [8]. They are leading to big data management problems, such as long-range communication capability and battery life catering for the off-grid IoT applications [9].

Edge computing has become an increasingly valued solution for reducing latency. Edge computing makes computation at the edge of the network where data are stored to improve data processing efficiency [10]. Edge computing is conceived as a promising solver for cloud computing problem [11], [12]. Recently, it is expected that a large amount of data will be generated, increasing the latency of traditional cloud computing [13]. In order to reduce the latency, it is better considering using an edge-computing architecture to offload some workloads from small devices to nearby edge servers with sufficient computing resources [14], [15]. The trend of using low-power wide-area network (LPWAN) communication technology in IoT, such as Sigofox, long range (LoRa), and weightless, has been widely applied in smart grid, smart city, and other fields [16]. Tens of thousands of sensors across a large area can be connected to this single gateway.

In order to build a smart edge-computing environment, this work uses Raspberry Pi 4 and Jetson Nano as edge devices. Also, it uses the message queuing telemetry transport (MQTT) transmission protocol to transmit data through the LoRa wide-area network (LoRaWAN) with long-distance, low-power transmission characteristics [17]. The proposed system collects air quality information from the Taiwan Environmental Protect Administration (EPA) stations, nongovernmental microsensors, AirBox, and the self-made sensors in Tunghai University campus. In order to manage such a large number of deployment devices, this study also demonstrates the concept of long-distance low-power edge-computing architecture with edge computing [18]–[20]. Also, an object detection on edge using Raspberry Pi and Intel neural compute stick (NCS) is demonstrated.

The objectives of this article are listed as follows.

1) Design and implement sensors, edge, and cloud (iSEC) framework.
2) Implement and evaluate the edge-computing performance of air quality data in the iSEC framework.

Manuscript received March 28, 2020; revised May 19, 2020; accepted June 16, 2020. Date of publication June 22, 2020; date of current version December 21, 2020. This work was supported in part by the Ministry of Science and Technology (MOST), Taiwan, under Grant 108-2221-E-029-010, Grant 108-2745-8-029-007, and Grant 108-2622-E-029-007-CC3; and in part by the National Applied Research Laboratories (NARLabs), Taiwan, under Grant 03108F1106. *(Corresponding author: Chao-Tung Yang.)*

Endah Kristiani is with the Department of Industrial Engineering and Enterprise Information, Tunghai University, Taichung 40704, Taiwan, and also with the Department of Informatics, Faculty of Engineering and Computer Science, Krida Wacana Christian University, Jakarta 11470, Indonesia.

Chin-Yin Huang is with the Department of Industrial Engineering and Enterprise Information, Tunghai University, Taichung 40704, Taiwan.

Chao-Tung Yang and Po-Cheng Ko are with the Department of Computer Science, Tunghai University, Taichung 40704, Taiwan (e-mail: ctyang@thu.edu.tw).

Halim Fathoni is with the Department of Industrial Engineering and Enterprise Information, Tunghai University, Taichung 40704, Taiwan, and also with the Department Ekonomi dan Bisnis, Polýteknýk Negeri, Lampung 35141, Indonesia.

Digital Object Identifier 10.1109/JIOT.2020.3004244

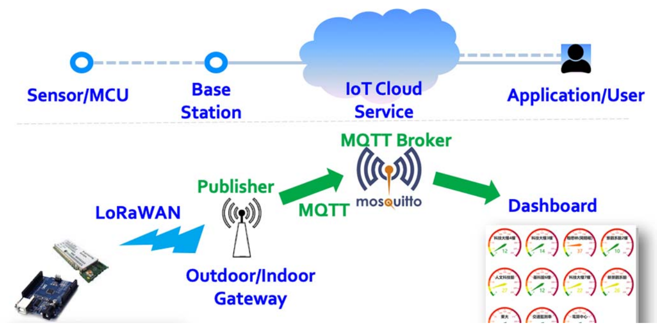

Fig. 1. MQTT data transmission.

3) Demonstrate and evaluate the edge-computing performance of the object detection case in the iSEC framework.

Therefore, the contribution of this article is to propose an integration of iSEC with two kinds of use cases, air quality data, and object detection implemented in the proposed system. The experiments and evaluation result might be a comparative study for developers in integrating sensor, edge, and cloud computing frameworks.

II. Background Review and Related Works

In this section, we provide several components that are utilized in this work: IoT, cloud computing, edge computing, MQTT, and data lake. The next sections discuss each component in more detail.

A. Internet of Things

The IoT is a device connected Internet that requires microsensing chips to be attached to specific objects: including radio frequency identification (RFID), sensors, and wireless communication chips. Communicate and dialogue (physical-to-object communication, object-to-person communication, and person-to-person communication) through ubiquitous devices and facilities through a variety of wireless and wired communication networks to provide management and service functions based on the semantic Web technology, to achieve the integration of "all things," efficient, power saving, security, and environment friendly.

B. Cloud Computing and Edge Computing

1) Cloud Computing: Cloud computing is a ubiquitous network that accesses the operating modes of shared computing resources (such as networks, servers, storage, applications, and services) in a convenient and on-demand manner. The computing resource is provided with minimal administrative work and publisher to configure and allocate computing resources quickly. Users no longer need to know the details of the infrastructure in the "cloud," they do not have to have the appropriate knowledge, and they do not require control directly. Cloud computing describes a new Internet-based delivery model that typically uses virtualized resources over the Internet.

2) Edge Computing: Edge computing is a concept of near computation (distributed computing architecture). It calculates the computation closer to the location of the resource. It does not transfer data back to the cloud and reduces the cost of data to and from the cloud. Edge computing is a difference with the cloud that the operation is moved to the edge node were the location for local processing [21]. Edge computing usually performs operations near the local and cloud interfaces, that is, the location of data in and out of the local area network, convenient for the simultaneous connection with the cloud and the local.

C. Message Queuing Telemetry Transport

Because the edge device does not necessarily have a good environment and power supply, in order to be able to transfer data and save energy quickly is also a major issue, the system will use MQTT as a bridge between machines. MQTT is a communication protocol for the IoT; IBM and Eurotech originally developed it. The purpose of MQTT is to provide a lightweight pipeline between the sensor and the satellite under the premise of narrow network bandwidth and low power consumption, and reliable communication protocols. Later in 2011, the MQTT agreement was donated to the Eclipse Foundation, which manages the development of source code projects, and officially became the OASIS international standard in 2014. MQTT is a message transmission protocol based on the "Publish/Subscribe" mechanism, as shown in Fig. 15. This mechanism is similar to the very popular YouTube model at the moment. The creator uploads the work to the YouTube platform (Publish) instead of sending it directly to the audience, and the audience then subscribes to the favorite content (Subscribe) according to their interests. The YouTube platform (broker) is responsible for unifying and managing all content, and then according to the content that each user subscribes to provides the audience. Fig. 1 shows the MQTT data transmission schema.

D. Cloud Edge Model With Data Lake

In the real world, machine-to-machine (M2M) communications need to use protocols before connection, common ones, such as HTTP, and MQTT is suitable for IoT. The reason is that it has a smaller volume packet, which means that it requires less transmission time and low power consumption. Compared to HTTP, the header of MQTT is only 2 B. In addition to the actual data to be transmitted in HTTP, the packet's header contains many users.

After we deploy the smart edge environment and use the MQTT protocol to send data to the cloud through LoRaWAN, we can import the data into the data lake. To effectively allocate the resources used to construct and manage the data lake, it is helpful to define each area and understand the relationship between the various parts. Once the data are stored, it will not change over time. There must be a time attribute. Generally, a data lake will contain a large amount of historical data. In the future, we can use specific analysis methods to dig out the data for visual presentation, as shown in Fig. 2.

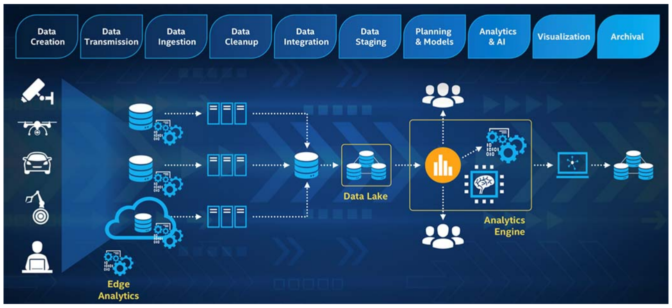

Fig. 2. Smart cloud edge model with data lake.

E. Related Works

With the rapid development of the IoT and 5G networks in smart urban environments, it is expected that a large amount of data will be generated, increasing the latency of traditional cloud computing. Wang *et al.* [14] studied the edge server layout problem in a smart city mobile-edge computing environment to reduce latency. The edge server is placed in a critical position to balance the workload of the edge server. Abdellatif *et al.* [5] proposed the usage of edge computing to monitor, process, and develop a vision for intelligent health application autonomous decision making. A selective data transmission scheme is implemented, which selects the most convenient data transmission method according to the detected condition of the patient. Hossain *et al.* [22] used edge calculation methods to minimize this delay. Since the main part of the data is generated from user endpoints, processing this data at the edge can significantly improve performance. This work deploys edge nodes at the sensor location to reduce latency.

Wireless sensor networks (WSNs) are facing lifespan challenges, packet loss during transmission, and end-to-end latency. The static assignment of sensor nodes and gateways applies to predefined and preplanned WSNs. The sensor nodes in this project need more gateway to complete the entire IoT scenario. Babazadeh [23] presented a project that developed on a single-core Intel Curie processor. Arduino101 and the raw sensor data are compressed using a customized Lempel–Ziv method. The compression rate figures are then analyzed; however, only the compressed data are sent back to the server through an LoRa platform. Roy *et al.* [24] proposed a gateway load balancing solution to overcome these problems. Our study refers to this work by using two LoRa gateways for sensor node data collection. Fog computing is expected to introduce cloud-like services on the local network while reducing costs. Nasir *et al.* [25] proposed a new resource-efficient framework for distributed video summarization on multiregion fog computation paradigms. The nodes of the Fog network are based on the resource-constrained device Raspberry Pi. The surveillance video is distributed across different nodes and periodically pushed to the cloud to reduce bandwidth consumption. This study uses the Raspberry Pi for object recognition at the edge. The experimental results show that even with minimal resources, the single-board computer requires only a small amount of overhead and has excellent scalability without using expensive cloud solutions to verify its presence.

Morabito *et al.* [26] introduced a lightweight edge gateway for the IoT (LEGIoT) architecture. It relies on the modular nature of microservices and the flexibility of the lightweight virtualization technology to ensure a scalable and flexible solution. Song *et al.* [27] proposed a new information infrastructure called the Energy IoT (IoET) to make DSM practical based on the latest wireless communication technology: LPWAN. The main advantage of LPWAN over general packet radio service (GPRS) and regional IoT is its wide-area coverage, which has the lowest power and maintenance costs. The monitoring network proposed by Addabbo *et al.* [28] consists of shallow power sensor nodes with LoRa connections capable of measuring the displacement of structural cracks in buildings with 10-μ resolution. Ponce and Gutiarrez [29] discussed an IoT system for predicting climatic conditions in closed areas using supervised learning methods, artificial hydrocarbon network models, using artificial intelligence (AI), over ten days of data. The experimental results conclude that the artificial hydrocarbon network model helps predict remote temperatures.

Yin *et al.* [11] proposed an advanced decision model to solve the computational unloading problem in edge computing. It uses the inherent hierarchical topology of the Internet to perform online scheduling in a decentralized manner to eliminate the expected modeling. Parallel message passing interface (MPI) and OpenMP parallel programming rely on the concept proposed by Yang *et al.* [30] in C language. Since the parallel loop self-scheduling consists of static and dynamic allocation, the static part adopts the weighting algorithm, and the dynamic part takes the famous cyclic self-scheduling. This study applies MPI for parallel data preprocessing.

Tiwary *et al.* [31] introduced a noncooperative broad game model in which players can maximize their rewards to minimize response time. The game model achieves a balance by using a reverse induction technique. This work also handles device availability by clustering previous availability data. Finally, the proposed model's performance is evaluated based on response time, user efficiency, and memory utilization.

III. System Architecture and Implementation

In this section, the development of the iSEC framework is presented. This project proposes a smart service architecture that uses LoRaWAN to build an intelligent IoT and edge cloud computing integration environment. Also, an optimized intelligent edge-computing architecture using deep learning models is presented. The system uses a Raspberry Pi on edge to serve the application. In this environment, device layers (such as sensors and cameras) will be connected to the Raspberry Pi embedded AI model to predict, detect, analyze, or identify patterns based on the object detection [32]. To achieve interconnection, we use MQTT broker as the machine-to-machine connection protocol. With this architecture, this system can implement smart system development in two scenarios, that is object detection and time-series cases. This project is part of the iSEC framework, which is divided into

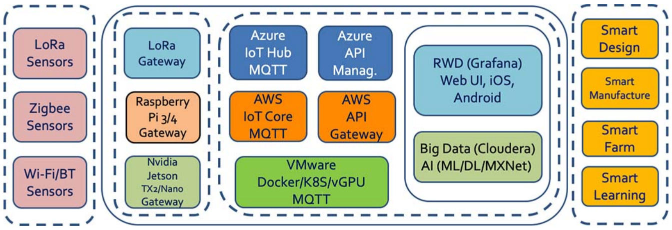

Fig. 3. iSEC framework.

three parts: 1) cloud; 2) edge; and 3) device. In cloud computing, the system environment consists of a private virtual machine (VM) and a provider's cloud, such as Amazon Web service (AWS) and Azure. In edge computing, the system environment consists of Raspberry Pi, Nvidia Jetson Nano, and LoRa gateway. The device section consists of several sensor gateways, such as LoRa, ZigBee, Wi-Fi/BT sensors, and cameras. On the device side, it consists of devices, such as sensors, Raspberry Pi, Arduino, Jetson Nano, and Linkit One. This platform is also applicable to various transmission protocols, such as ZigBee, MQTT, Wi-Fi, NB-IoT, and Bluetooth. Through LoRaWAN, the sensing data collected by LoRaWAN nodes is transmitted to the LoRaWAN gateway. During this process, LoRaWAN provides an LPWAN and a star topology. The gateway is connected to the microedge ramp made by Raspberry Pi, and it is finally uniformly transmitted to the data center. Fig. 3 describes the entire of the development framework.

A. Cloud System Architecture

The data lake uses Hadoop as a storage and computing platform for big data. It then uses HBase as a database for data storage. Also, it uses Spark MLlib to perform data exploration, clustering, classification, and regression algorithms. Other methods are to find a large correlation amount of data and turn the data into the required information. For future development, the data will accumulate and store in a large number of data sets with a variety of criteria, such as the length of time, the increasing, and decreasing of data sources. This historical data will process and provide intelligent systems, such as to predict future data. The data lake storage system will use a multisupported and scalable Linux as the operating system. Hadoop distributed file system (HDFS) is applied in the Hadoop computing architecture to perform overall data storage. HDFS has high fault tolerance and excellent stability. Therefore, this project uses HDFS as an essential storage system, as shown in Fig. 4.

B. LoRaWAN Architecture

LoRaWAN has a long-distance and low-power wireless network technology. Through this technology, we can build a low power wide area (LPWA). The low-power characteristics make IoT applications have excellent battery life. The advantage of long distance can enable a single gateway or base station to cover the entire city or hundreds of square kilometers. Therefore, the system can be expanded in the future. Fig. 5 shows the architecture of LoRaWAN connection.

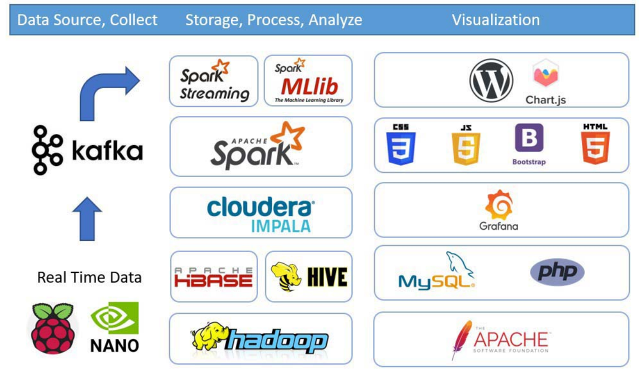

Fig. 4. Cloud system architecture.

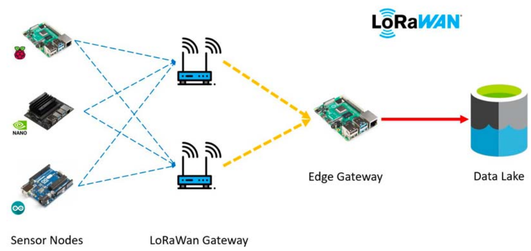

Fig. 5. LoRaWAN system architecture.

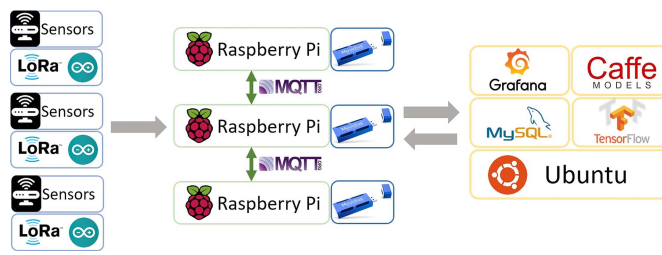

Fig. 6. Edge system architecture.

C. Edge System Architecture

In order to build a smart edge-computing environment, this system uses Raspberry Pi 4 and Jetson Nano as an edge device. The edge-computing layer implements on two cases of time-series data and object detection. The time-series case was applied to the air quality data of Taiwan, while the object detection case was applied using an available model. The edge system architecture on Raspberry Pi is shown in Fig. 6.

1) Air Quality Data Set: The system uses the edge-computing architecture to perform data acquisition and data preprocessing on official stations, folk airbox, and self-made sensors at the Tunghai campus. Also, to collect the air quality data from various sources in asynchronous and real time. Using a python crawler on the Raspberry Pi, the processed data are sent to the MySQL relational database for storage, providing a follow-up prediction application. The MQTT is used to communicate with the machine and machine between the edge nodes and alert the abnormal information.

There are three types of air quality data need to be collected.

1) Taiwan Environmental Protect on Administration 77 national stations and 740 microsensors.
2) Taiwan Academia Sinica (AS) 2139 microsensors.

```
[{"SiteName":"屏東(琉球)","County":"屏東縣","AQI":"56","Pollutant":"細懸浮微粒","Status":"普
通","SO2":"","CO":"","CO_8hr":"0.3","O3":"33","O3_8hr":"20","PM10":"","PM2.5":"","NO2":"2.6","NOx":"3.3","NO":"0.6","WindSpeed":"4.5","WindDirec":"3
18","PublishTime":"2019-04-10 11:00","PM2.5_AVG":"18","PM10_AVG":"24","SO2_AVG":"2","Longitude":"120.377222","Latitude":"22.352222"},{"SiteName":"苗
栗(後龍)","County":"苗栗縣","AQI":"63","Pollutant":"細懸浮微粒","Status":"普
通","SO2":"0.9","CO":"0.19","CO_8hr":"0.3","O3":"34","O3_8hr":"17","PM10":"41","PM2.5":"18","NO2":"4.1","NOx":"5.4","NO":"1.3","WindSpeed":"5.6","Wi
ndDirec":"218","PublishTime":"2019-04-10 11:00","PM2.5_AVG":"20","PM10_AVG":"37","SO2_AVG":"2","Longitude":"120.786028","Latitude":"24.616369"},
{"SiteName":"彰化(大城)","County":"彰化縣","AQI":"54","Pollutant":"細懸浮微粒","Status":"普
通","SO2":"2.1","CO":"0.2","CO_8hr":"0.2","O3":"30","O3_8hr":"19","PM10":"25","PM2.5":"16","NO2":"3.6","NOx":"5.7","NO":"2.1","WindSpeed":"3.3","Win
dDirec":"251","PublishTime":"2019-04-10 11:00","PM2.5_AVG":"17","PM10_AVG":"28","SO2_AVG":"1","Longitude":"120.273117","Latitude":"23.843139"},
{"SiteName":"臺南(北門)","County":"臺南市","AQI":"37","Pollutant":"","Status":"良
好","SO2":"2.8","CO":"0.09","CO_8hr":"0.1","O3":"33","O3_8hr":"21","PM10":"27","PM2.5":"14","NO2":"2.2","NOx":"2.9","NO":"0.7","WindSpeed":"3.6","Wi
ndDirec":"242","PublishTime":"2019-04-10 11:00","PM2.5_AVG":"11","PM10_AVG":"29","SO2_AVG":"1","Longitude":"120.124167","Latitude":"23.265066"},
{"SiteName":"富貴角","County":"新北市","AQI":"64","Pollutant":"細懸浮微粒","Status":"普
通","SO2":"0.3","CO":"0.16","CO_8hr":"0.2","O3":"33","O3_8hr":"17","PM10":"51","PM2.5":"21","NO2":"1.3","NOx":"1.7","NO":"0.4","WindSpeed":"6.6","Wi
ndDirec":"252","PublishTime":"2019-04-10 11:00","PM2.5_AVG":"21","PM10_AVG":"50","SO2_AVG":"1","Longitude":"121.536763","Latitude":"25.298562"},
{"SiteName":"麥寮","County":"雲林縣","AQI":"41","Pollutant":"","Status":"良
好","SO2":"0.5","CO":"0.21","CO_8hr":"0.3","O3":"28","O3_8hr":"16","PM10":"34","PM2.5":"15","NO2":"2.2","NOx":"3","NO":"0.8","WindSpeed":"4.4","Wind
Direc":"230","PublishTime":"2019-04-10 11:00","PM2.5_AVG":"13","PM10_AVG":"37","SO2_AVG":"1","Longitude":"120.251825","Latitude":"23.753506"},
{"SiteName":"關山","County":"臺東縣","AQI":"45","Pollutant":"","Status":"良好","SO2":"1.4","CO":"-
","CO_8hr":"","O3":"33","O3_8hr":"17","PM10":"35","PM2.5":"13","NO2":"3.7","NOx":"4.9","NO":"1.3","WindSpeed":"1.9","WindDirec":"204","PublishTime":
"2019-04-10 11:00","PM2.5_AVG":"14","PM10_AVG":"31","SO2_AVG":"1","Longitude":"121.161933","Latitude":"23.045083"},{"SiteName":"馬公","County":"澎湖
縣","AQI":"59","Pollutant":"細懸浮微粒","Status":"普
通","SO2":"1.3","CO":"0.27","CO_8hr":"0.3","O3":"29","O3_8hr":"32","PM10":"","PM2.5":"","NO2":"2.5","NOx":"6.2","NO":"3.7","WindSpeed":"2.3","WindDi
rec":"241","PublishTime":"2019-04-10 11:00","PM2.5_AVG":"19","PM10_AVG":"30","SO2_AVG":"1","Longitude":"119.566158","Latitude":"23.569031"},
{"SiteName":"金門","County":"金門縣","AQI":"71","Pollutant":"細懸浮微粒","Status":"普
通","SO2":"3.3","CO":"0.3","CO_8hr":"0.3","O3":"18","O3_8hr":"19","PM10":"58","PM2.5":"33","NO2":"11","NOx":"17","NO":"6.2","WindSpeed":"3.7","WindD
irec":"191","PublishTime":"2019-04-10 11:00","PM2.5_AVG":"24","PM10_AVG":"47","SO2_AVG":"2","Longitude":"118.312256","Latitude":"24.432133"},
{"SiteName":"馬祖","County":"連江縣","AQI":"65","Pollutant":"細懸浮微粒","Status":"普
通","SO2":"1.9","CO":"0.34","CO_8hr":"0.3","O3":"32","O3_8hr":"23","PM10":"53","PM2.5":"22","NO2":"6.3","NOx":"7.4","NO":"1.1","WindSpeed":"3.6","Wi
ndDirec":"327","PublishTime":"2019-04-10 11:00","PM2.5_AVG":"21","PM10_AVG":"48","SO2_AVG":"2","Longitude":"119.949875","Latitude":"26.160469"},
{"SiteName":"埔里","County":"南投縣","AQI":"52","Pollutant":"細懸浮微粒","Status":"普
```


Fig. 7. Raw data of EPA stations.


```
{"message":"Success","status":200,"data":[{"updatetime":"2019-04-10T11:00:00","deviceId":"7526032148","pm2_5":12},
{"updatetime":"2019-04-10T11:00:00","deviceId":"7523264126","pm2_5":null},{"updatetime":"2019-04-
10T11:00:00","deviceId":"7524131114","pm2_5":13},{"updatetime":"2019-04-10T11:00:00","deviceId":"7527459181","pm2_5":15},
{"updatetime":"2019-04-10T11:00:00","deviceId":"7527563599","pm2_5":14},{"updatetime":"2019-04-
10T11:00:00","deviceId":"7524224279","pm2_5":null},{"updatetime":"2019-04-10T11:00:00","deviceId":"7515331409","pm2_5":null},
{"updatetime":"2019-04-10T11:00:00","deviceId":"7513785359","pm2_5":null},{"updatetime":"2019-04-
10T11:00:00","deviceId":"7495274841","pm2_5":null},{"updatetime":"2019-04-10T11:00:00","deviceId":"7492132843","pm2_5":null},
{"updatetime":"2019-04-10T11:00:00","deviceId":"7526625381","pm2_5":null},{"updatetime":"2019-04-
10T11:00:00","deviceId":"7494879805","pm2_5":null},{"updatetime":"2019-04-10T11:00:00","deviceId":"7524001112","pm2_5":null},
{"updatetime":"2019-04-10T11:00:00","deviceId":"7527351738","pm2_5":null},{"updatetime":"2019-04-
10T11:00:00","deviceId":"7494269836","pm2_5":null},{"updatetime":"2019-04-10T11:00:00","deviceId":"7495775458","pm2_5":null},
{"updatetime":"2019-04-10T11:00:00","deviceId":"7514167092","pm2_5":null},{"updatetime":"2019-04-
10T11:00:00","deviceId":"7523910079","pm2_5":null},{"updatetime":"2019-04-10T11:00:00","deviceId":"7492215675","pm2_5":null},
{"updatetime":"2019-04-10T11:00:00","deviceId":"7522721084","pm2_5":null},{"updatetime":"2019-04-
10T11:00:00","deviceId":"7495890482","pm2_5":null},{"updatetime":"2019-04-10T11:00:00","deviceId":"7523081612","pm2_5":null},
{"updatetime":"2019-04-10T11:00:00","deviceId":"7492766302","pm2_5":null},{"updatetime":"2019-04-
10T11:00:00","deviceId":"7493298792","pm2_5":null},{"updatetime":"2019-04-10T11:00:00","deviceId":"7513366876","pm2_5":null},
{"updatetime":"2019-04-10T11:00:00","deviceId":"7524334637","pm2_5":null},{"updatetime":"2019-04-
10T11:00:00","deviceId":"7523746557","pm2_5":null},{"updatetime":"2019-04-10T11:00:00","deviceId":"7501204506","pm2_5":null},
{"updatetime":"2019-04-10T11:00:00","deviceId":"7478714726","pm2_5":null},{"updatetime":"2019-04-
10T11:00:00","deviceId":"7523404601","pm2_5":null},{"updatetime":"2019-04-10T11:00:00","deviceId":"7527014441","pm2_5":null},
{"updatetime":"2019-04-10T11:00:00","deviceId":"7492511830","pm2_5":null},{"updatetime":"2019-04-
10T11:00:00","deviceId":"7494636687","pm2_5":null},{"updatetime":"2019-04-10T11:00:00","deviceId":"7494072208","pm2_5":null},
{"updatetime":"2019-04-10T11:00:00","deviceId":"7492877569","pm2_5":null},{"updatetime":"2019-04-
10T11:00:00","deviceId":"7495672573","pm2_5":null},{"updatetime":"2019-04-10T11:00:00","deviceId":"7522385911","pm2_5":null},
{"updatetime":"2019-04-10T11:00:00","deviceId":"7522863555","pm2_5":null},{"updatetime":"2019-04-
10T11:00:00","deviceId":"7494165695","pm2_5":null},{"updatetime":"2019-04-10T11:00:00","deviceId":"7523357297","pm2_5":null},
```


Fig. 8. Raw data of EPA microsensors.


```
{"source": "last-all-airbox by IIS-NRL", "feeds": [{"gps_num": 9.0, "app": "AirBox", "s_d1": 50.0, "fmt_opt": "1", "s_d2":
23.0, "c_d0_method": "LR/7/2.06", "s_d0": 39.0, "gps_alt": 2.0, "s_h0": 66.0, "SiteName": "74DA38F7C368", "gps_fix": 1.0,
"ver_app": "1.0.7", "gps_lat": 24.999, "s_t0": 32.25, "timestamp": "2019-04-10T02:30:56Z", "gps_lon": 121.521, "date": "2019-
04-10", "tick": 1554863456.0, "device_id": "74DA38F7C368", "s_1": 100.0, "s_0": 457.0, "s_3": 0.0, "s_2": 1.0, "ver_format":
"3", "c_d0": 48.07, "time": "02:30:56"}, {"gps_num": 9.0, "app": "AirBox", "s_d1": 34.0, "fmt_opt": "1", "s_d2": 18.0, "s_d0":
31.0, "gps_alt": 2.0, "s_h0": 79.0, "SiteName": "\u82d7\u6817\u7e23\u7e23\u7acb\u5357\u57d4\u570b\u5c0f", "gps_fix": 1.0,
"ver_app": "1.0.7", "gps_lat": 24.635, "s_t0": 31.62, "timestamp": "2019-04-10T01:52:41Z", "gps_lon": 120.973, "date": "2019-
04-10", "tick": 1554861161.0, "device_id": "74DA38F20A3A", "s_1": 100.0, "s_0": 348.0, "s_3": 0.0, "s_2": 1.0, "ver_format":
"3", "time": "01:52:41"}, {"gps_num": 9.0, "app": "AirBox", "s_d1": 36.0, "fmt_opt": "1", "s_d2": 20.0, "s_d0": 32.0,
"gps_alt": 2.0, "s_h0": 72.0, "SiteName": "\u82d7\u6817\u7e23\u7e23\u7acb\u7530\u7f8e\u570b\u5c0f", "gps_fix": 1.0, "ver_app":
"1.0.7", "gps_lat": 24.629, "s_t0": 30.37, "timestamp": "2019-04-10T01:45:11Z", "gps_lon": 121.008, "date": "2019-04-10",
"tick": 1554860711.0, "device_id": "74DA38F20A3C", "s_1": 100.0, "s_0": 537.0, "s_3": 0.0, "s_2": 1.0, "ver_format": "3",
"time": "01:45:11"}, {"gps_num": 9.0, "app": "AirBox", "s_d1": 55.0, "fmt_opt": "1", "s_d2": 27.0, "s_d0": 45.0, "gps_alt":
2.0, "s_h0": 69.0, "SiteName": "\u82b1\u84ee\u7e23\u7acb\u660e\u5ec9\u570b\u5c0f", "gps_fix": 1.0, "ver_app": "1.0.7",
"gps_lat": 23.994, "s_t0": 32.0, "timestamp": "2019-04-10T02:33:20Z", "gps_lon": 121.594, "date": "2019-04-10", "tick":
1554863600.0, "device_id": "74DA38ED0630", "s_1": 100.0, "s_0": 1830.0, "s_3": 0.0, "s_2": 1.0, "ver_format": "3", "time":
"02:33:20"}, {"gps_num": 9.0, "app": "AirBox", "s_d1": 27.0, "fmt_opt": "1", "s_d2": 19.0, "s_d0": 26.0, "gps_alt": 2.0,
"s_h0": 58.0, "SiteName": "74DA38EC5734", "gps_fix": 1.0, "ver_app": "1.0.7", "gps_lat": 13.168, "s_t0": 36.87, "timestamp":
"2019-04-10T02:40:26Z", "gps_lon": 100.927, "date": "2019-04-10", "tick": 1554864026.0, "device_id": "74DA38EC5734", "s_1":
100.0, "s_0": 1159.0, "s_3": 0.0, "s_2": 1.0, "ver_format": "3", "time": "02:40:26"}, {"gps_num": 9.0, "app": "AirBox",
"s_d1": 45.0, "fmt_opt": "1", "s_d2": 25.0, "c_d0_method": "LR/7/3.04", "s_d0": 36.0, "gps_alt": 2.0, "s_h0": 87.0,
"SiteName": "74DA38B0538E", "gps_fix": 1.0, "ver_app": "0.35.2", "gps_lat": 25.143, "s_t0": 33.5, "timestamp": "2019-04-
10T02:38:24Z", "gps_lon": 121.734, "date": "2019-04-10", "tick": 1554863904.0, "device_id": "74DA38B0538E", "s_1": 100.0,
"s_0": 10.0, "s_3": 0.0, "s_2": 1.0, "ver_format": "3", "c_d0": 16.48, "time": "02:38:24"}, {"gps_num": 9.0, "app": "AirBox",
"s_d1": 48.0, "fmt_opt": "1", "s_d2": 27.0, "s_d0": 38.0, "gps_alt": 2.0, "s_h0": 74.0, "SiteName": "74DA38B0538C", "gps_fix":
1.0, "ver_app": "1.0.7", "gps_lat": 0.0, "s_t0": 32.0, "timestamp": "2019-04-10T01:08:07Z", "gps_lon": 0.0, "date": "2019-04-
10", "tick": 1554858487.0, "device_id": "74DA38B0538C", "s_1": 100.0, "s_0": 80.0, "s_3": 0.0, "s_2": 1.0, "ver_format": "3",
```


Fig. 9. Raw data of LASS microsensors.

3) The data from the Tunghai campus crawled by the edge node.

2) Retrieve Data of Station: Taiwan Environmental Protect on Administration of 77 national stations.

The application program interface (API) format by EPA is shown in Fig. 7.

Taiwan Environmental Protect on Administration of 740 microsensors.

The format of API by EPA is shown in Fig. 8.

Taiwan AS 2139 microsensors.

The API format by AS is shown in Fig. 9.

3) Retrieve Data of Self-Made Sensor: This study integrated the Arduino UNO board, PMS5003T sensor, and LoRa module to get the $PM_{2.5}$, PM_{10}, temperature, and humidity data. The LoRa node and LoRaWAN gateway are shown in Fig. 10. LoRa gateway configuration is shown in Fig. 11.

There are eight LoRa nodes in the Tunghai campus, and they need to be crawled one by one in the XML format, as shown in Fig. 12.

Fig. 10. LoRa node and LoRaWAN gateway.

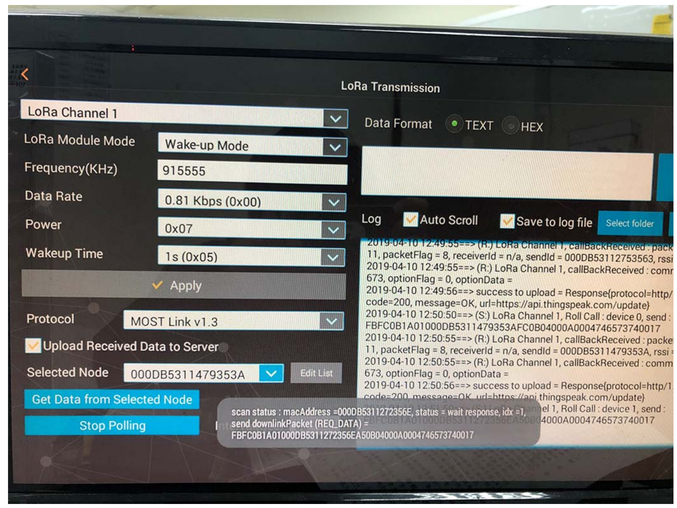

Fig. 11. LoRa gateway network configuration.

This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
▾<feed>
  <created-at type="dateTime">2019-04-10T11:54:45+08:00</created-at>
  <entry-id type="integer">97605</entry-id>
  <field1>37.0</field1>
  <field2>22.0</field2>
  <field3>22.0</field3>
  <field4>23.0</field4>
</feed>
```


Fig. 12. Raw data of LoRa sensors.

Data Preprocessing is processed on the edge-computing side. Sometimes, the sensor is inevitably unstable; therefore, unreasonable extreme data caused. The reasonable range of each data is: temperature: 0 °C–40 °C, humidity: 0%–100%, $PM_{2.5}$: 0–500 ppm, and PM_{10}: 0–604 ppm. In order to solve this problem, this study preprocesses the data while retrieving the data and replacing the unreasonable data with null value. Since the frequency of the self-made sensor data is one record every 10 min, the percentage of missing data is <1%, which is ignored in this study.

4) MQTT: This study also applies the MQTT protocol to send and receive an alarm message for the user. LoRa sensor can detect the temperature based on the setting parameter. When the room temperature is set to 28 °C threshold in summer, it can infer someone opens the air conditioner. At the same time, the MQTT will publish a message to inform somebody in the room. The publisher and subscriber are shown in Fig. 13.

Moreover, this study deploys the Grafana dashboard on the raspberry pi for data visualization and historical data tracking, as shown in Fig. 14. This study applies the simple mail transfer

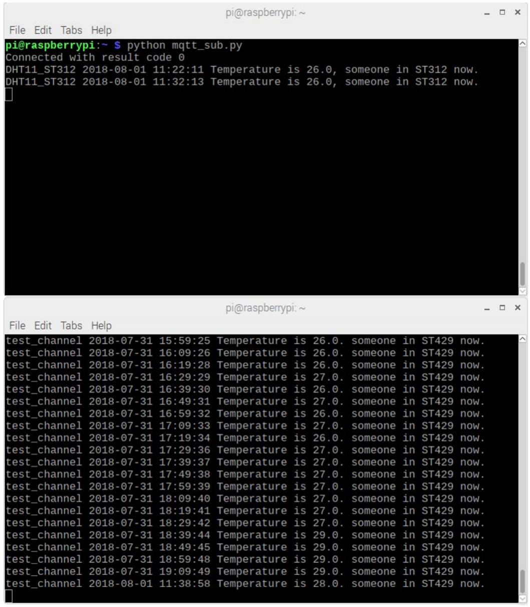

Fig. 13. RPi side publisher and subscriber.

Fig. 14. Grafana dashboard.

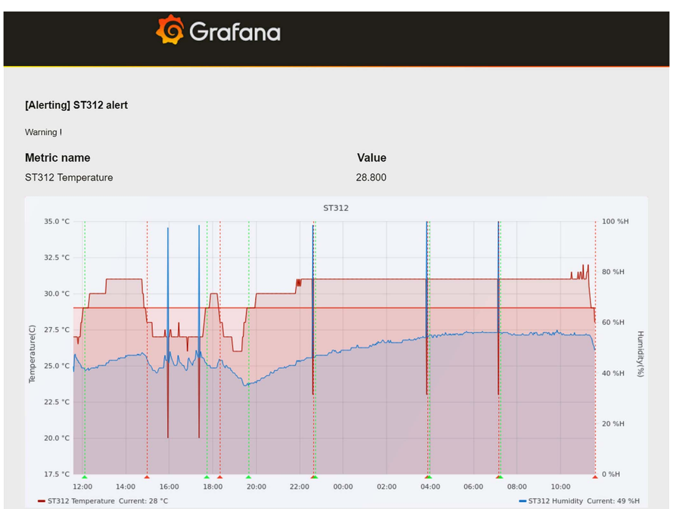

Fig. 15. Grafana alarm.

protocol (SMTP) to send an alarm message through e-mail. Once receiving the abnormal data, Grafana will capture the screenshot photograph and mail to a particular account, as shown in Fig. 15.

Fig. 16. Raspberry Pi with camera and NCS.

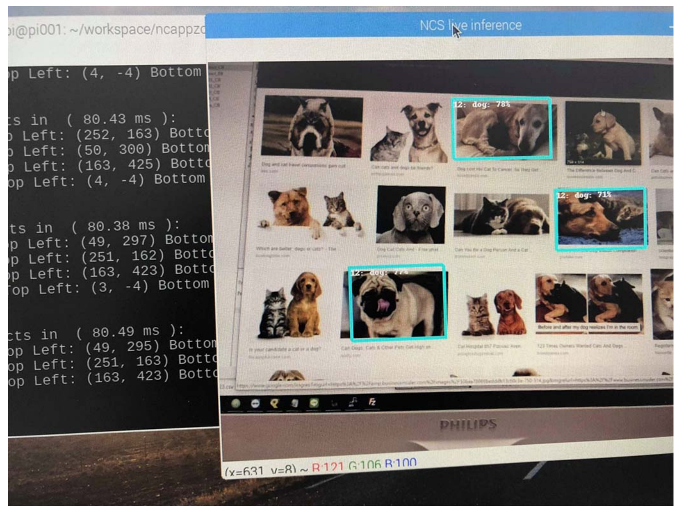

Fig. 17. NCS object detection.

5) *Object Detection Using NCS:* This study also integrated Raspberry Pi 3 B+ with the camera module and NCS for object detection, as shown in Fig. 16.

First, we installed the NCS software development kit (SDK) Tools on Ubuntu Desktop, and convert the Caffe model into the graph file that can be executed at the edge node through the SDK Tool. In this case, we only need to install the NCS SDK API on the Raspberry Pi and put the graph file generated on the Raspberry Pi. The Raspberry Pi has a deep learning ability to do object detection, as shown in Fig. 17.

IV. Experimental Results and Discussion

A. Hardware Environment

From the device side, this study deploys eight LoRa nodes and two LoRa gateways in the Tunghai campus. Two of the nodes are located in the indoor room, and the rest is placed outdoor; the specification of the LoRa node and LoRa gateway, as shown in Table I.

Raspberry Pi 3 model B+ was implemented as an edge device part. The specification of Raspberry Pi is shown in Table II.

TABLE I
LoRa Node and LoRa Gateway

LoRa Node	LoRa Gateway
LoRa Module : GlobalSat LM-130	LoRa Module : GlobalSat LM-230
Arduino Uno	WiFi: Dual Band 2.4GHz/5GHz 802.11 a/b/g/n
PMS5003T	Ethernet: Gigabit Ethernet(RJ45)
	Internal Storage : 4GB
	CPU : ARM CORTEX A9 Dual Core (Std. 850MHz)

TABLE II
Raspberry Pi Specification

Item	Description
Model	Raspberry Pi 3 B+
Operate System	Raspian Strtch
CPU	Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
RAM	1GB LPDDR2 SDRAM
Storage	MicroSD
Network	Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
WiFi	2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN
Bluetooth	Bluetooth 4.2, BLE
Power	5V/2.5A DC power input

TABLE III
LoRa Sensors Location

Device ID	lon	lat
Gateway A	120.596892	24.181436
A	120.596903	24.181424
B1	120.59556	24.177561
C1	120.59715	24.18135
D	120.596797	24.181062
Gateway B	120.597243	24.178692
C2	120.59782	24.17991
C3	120.59734	24.17886
C4	120.59741	24.17854
D6	120.599059	24.177561

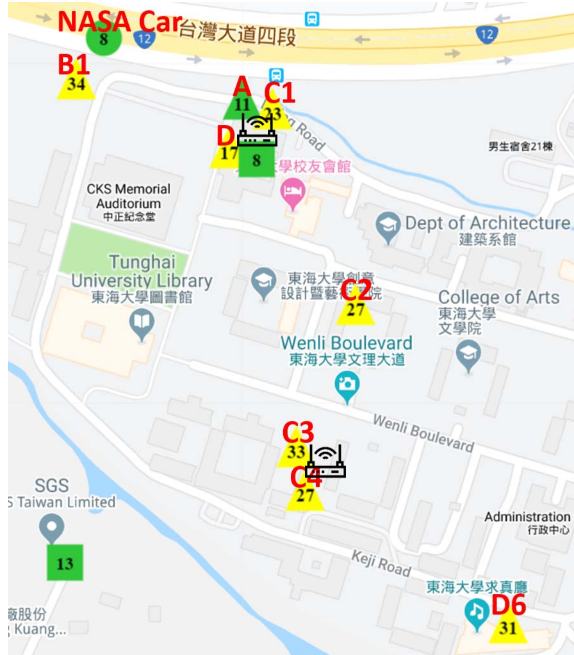

Fig. 18. LoRa sensors on campus.

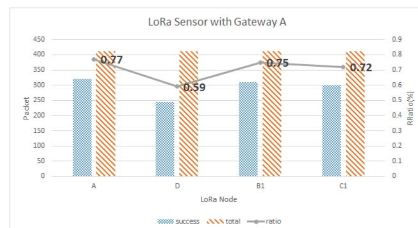

Fig. 19. LoRa package successful ratio of gateway A.

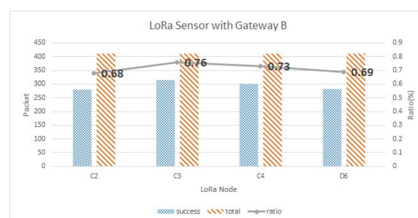

Fig. 20. LoRa package successful ratio of gateway B.

B. Experimental Results

This work sets up two LoRa gateways and eight LoRa nodes on campus to collect the school's air quality data.

1) Location of LoRa Nodes and Gateways: The coordinates are presented in Table III. The place is on the Tunghai campus, as shown in Fig. 18. We measured the success rate of LoRa packet transmission through the LoRa gateway A and LoRa gateway B. The result is shown in Figs. 19 and 20. It can be observed from the experiment that the transmission success rate is above 70%. Only the D sensor's success rate is less than 70% from LoRa gateway A. It is because of Gateway A, and D sensor are placed in a different floor, that could be more buildings block it.

2) Message Passing Interface Implementation: An MPI was installed on the Raspberry Pi to connect multiple Raspberry Pi to form an edge-computing node cluster to improve the computational efficiency of the single-board computer.

It is used to preprocess the data collected and test the execution time and performance under the different number of processes for different parameters, as shown in Figs. 21 and 22.

3) Data Normalization Processing: This study normalizes the data of every sensor at the campus by applying MPI on the edge nodes. Then, we also evaluate the network delay between cloud and edge, as shown in Fig. 23. Due to normalization requires a lot of and repeated operations, compared with all the calculations done by the cloud, the MPI can achieve relatively better time consumption performance at the edge nodes, as shown in Fig. 24.

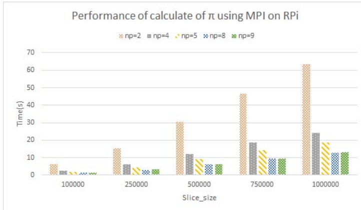

Fig. 21. MPI testing with different slice size.

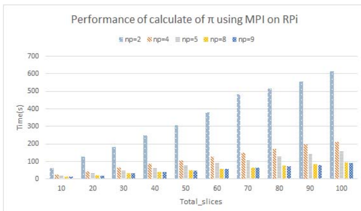

Fig. 22. MPI testing with different total slices.

Fig. 23. Network delay with ping command.

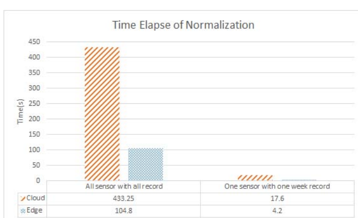

Fig. 24. Time elapse on normalization with MPI.

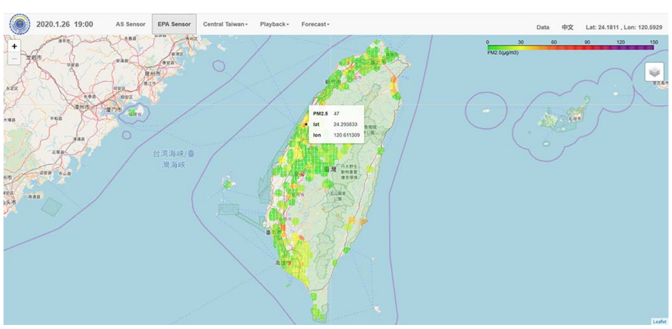

Fig. 25. PM2.5 status of EPA microsensors.

4) *Visualization:* There are three kinds of Web visualization in the experiments. First, from EPA consists of 77 national stations and 740 microsensors as demonstrated in Fig. 25. Second, from AS consists of 2139 microsensors as presented in Fig. 26. Third, the data of Tunghai campus crawled by the edge node as shown in Fig. 27.

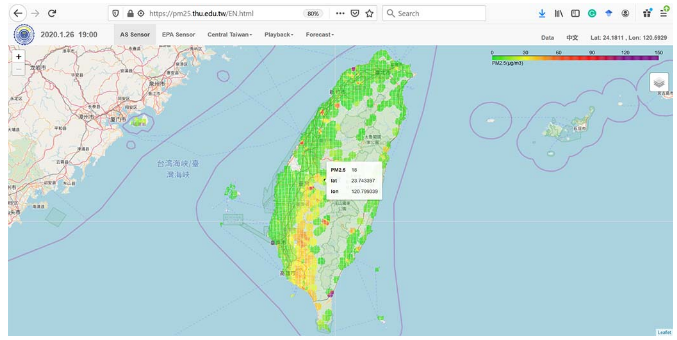

Fig. 26. PM2.5 status of AS microsensors.

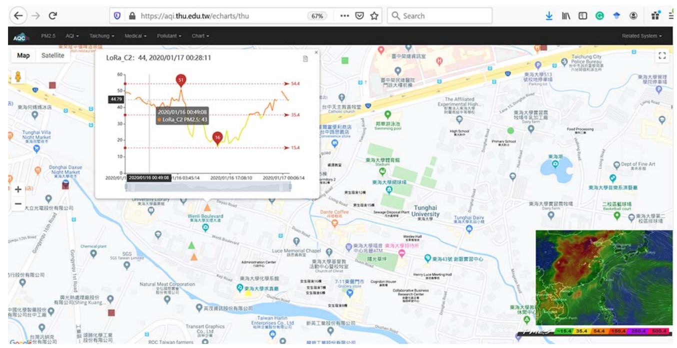

Fig. 27. PM2.5 status of Tunghai campus sensors.

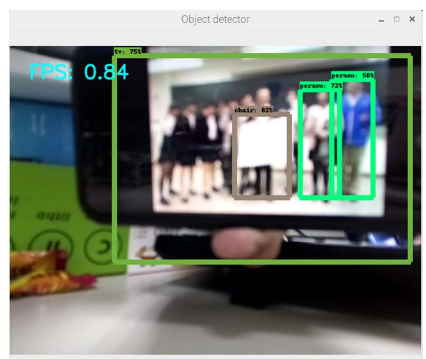

Fig. 28. Object detection without NCS.

5) *Object Detection on the Edge Node:* For object recognition, this study uses the NCS to enhance the computing power of the Raspberry Pi in processing the computer vision with NCS SDK to load the graph file and call SDK API by an edge node, speed up the recognition efficiency of the Raspberry Pi, and obtain nearly four times upgrades without NCS in frames per second (FPS).

The FPS without NCS reaches about 0.8 FPS, as shown in Fig. 28. The FPS with NCS reaches about 3.6 FPS, as shown in Fig. 29. This study also uses the power distribution unit (PDU) to record the power consumption w/ and w/o NCS. On average, it consumed 18.5 W during the execution of the object recognition without NCS and consumed 15.47 W with NCS. NCS saved roughly 16.37% energy consumption. The power consumption depicts on Fig. 30.

Fig. 29. Object detection with NCS.

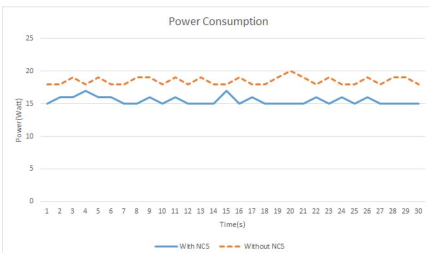

Fig. 30. Power consumption during object detection.

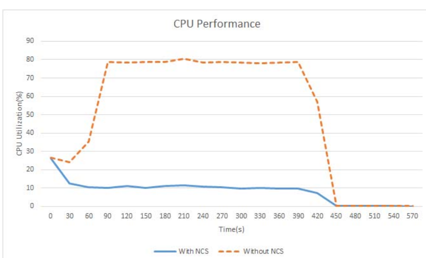

Fig. 31. CPU utility during object detection.

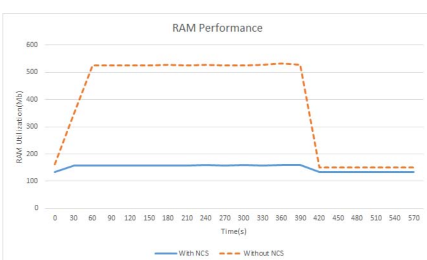

Fig. 32. Memory utility during object detection.

6) Resource Performance: This study also records the CPU, RAM, and temperature utilization during the program execution. In terms of CPU usage, a reduction of 82.96% was achieved, as shown in Fig. 31. In terms of RAM, it is 66.67% reduction, as shown in Fig. 32. The average temperature also reduced by roughly 8.8 °C, as shown in Fig. 33.

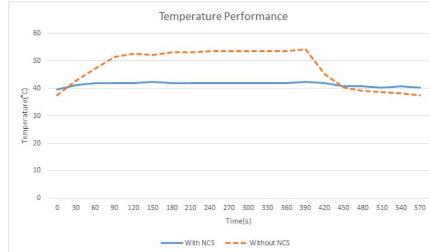

Fig. 33. Temperature during object detection.

Fig. 34. Jetson NANO performance.

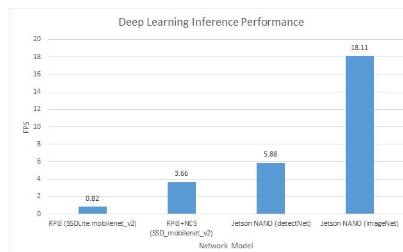

Fig. 35. FPS comparison.

TABLE IV
PERFORMANCE COMPARISON OF RASPBERRY PI AND JETSON NANO

	Raspberry Pi without NCS	**Raspberry Pi with NCS**	**Jetson Nano**
Power Consumption	18-20 watt	15-17 watt	10-11 watt
CPU	80%	10%	20%
RAM	520 Mb	160 Mb	3100 Mb
Temperature	54° C	42° C	40 – 70° C

In addition, this study do some experiments on the NVIDIA Jetson NANO, as shown in Fig. 34.

Finally, this work summarizes the performance of object detection using different network inference models at different edge nodes, which are Raspberry Pi and Jetson Nano, recording the average FPS, as shown in Fig. 35.

C. Discussion

Raspberry Pi and Jetson Nano are promising as the edge-computing devices. From the experiments, it can be seen in Table IV that Jetson Nano and Raspberry Pi have their characteristics. It all depends on the specific preferences.

V. Conclusion and Future Works

This article implemented iSEC, smart system integration of the sensors, cloud, and edge architecture. This system also performs and evaluates the edge-computing performance in the air quality data and object detection case. From the experiments, it can be seen that a combination of cloud edge computing provides a better service. A long-range, LPWAN communication module was applied to provide air quality monitoring sensors. It can be inferred that the data transmission about a 70% success rate in LoRa data transmission. The correlation analysis between the self-made sensor and the official station is roughly consistent with the trend. In terms of low cost, the self-made sensor is quite good. The Raspberry Pi was used as an edge-computing node for data preprocessing to reduce the load on the cloud server. The usage of the NCS to implement object detection on the Raspberry Pi also achieved more good performance and energy consumption and delivered a four times improvement in FPS. The FPS without NCS reaches about 0.8 FPS, and the FPS with NCS enters about 3.6 FPS. This study also uses the PDU to record the power consumption with and without NCS. On average, it consumed 18.5 W during the execution of object recognition without NCS. With NCS, it consumed 15.47 W. NCS saved roughly 16.37% energy consumption. In terms of CPU usage, a reduction of 82.96% was achieved. In terms of RAM, it is 66.67% reduction, and the average temperature is also reduced by roughly 8.8 °C. Raspberry Pi and Jetson Nano have their own characteristics. In Jetson Nano, the power consumption is around 10–11 W during the inference process. The CPU is at 20% usage from the total resource. The memory reaches at around 3100 Mb consumption, while the temperature is increasing from 40 °C–70 °C during the inference time.

In the future, this study will further strengthen the edge-computing side, such as expanding the number of edge nodes or replace the edge node. This study also increasing the number of campus sensors and plans to apply the outdoor LoRa gateway to achieve better data transmission quality. Therefore, it can collect more comprehensive data analysis by collecting sensor data distributed on campus. Then, we will use ML and deep learning techniques to conduct more comprehensive monitoring and forecasting of campus air quality. Also, do more detailed energy consumption experiments to get efficient power usage between edge computing and the cloud.

Acknowledgment

The authors are grateful to the National Center for High-performance Computing for computer resources and facilities.

References

[1] C.-T. Yang, Y.-W. Chan, J.-C. Liu, and B.-S. Lou, "An implementation of cloud-based platform with R packages for spatiotemporal analysis of air pollution," *J. Supercomput.*, vol. 76, no. 3, pp. 1416–1437, 2020.

[2] C.-T. Yang, C.-J. Chen, Y.-T. Tsan, P.-Y. Liu, Y.-W. Chan, and W.-C. Chan, "An implementation of real-time air quality and influenza-like illness data storage and processing platform," *Comput. Human Behav.*, vol. 100, pp. 266–274, Nov. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S074756321830493X

[3] C.-T. Yang, J.-C. Liu, S.-T. Chen, and H.-W. Lu, "Implementation of a big data accessing and processing platform for medical records in cloud," *J. Med. Syst.*, vol. 41, no. 10, pp. 127–136, 2017.

[4] M. Wu, N. N. Xiong, and L. Tan, "An intelligent adaptive algorithm for environment parameter estimation in smart cities," *IEEE Access*, vol. 6, pp. 23325–23337, 2018.

[5] A. A. Abdellatif, A. Emam, C.-F. Chiasserini, A. Mohamed, A. Jaoua, and R. Ward, "Edge-based compression and classification for smart healthcare systems: Concept, implementation and evaluation," *Expert Syst. Appl.*, vol. 117, pp. 1–14, Mar. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417418305967

[6] C.-K. Tsung, C.-T. Yang, and S.-W. Yang, "Visualizing potential transportation demand from ETC log analysis using ELK stack," *IEEE Internet Things J.*, early access, Feb. 18, 2020, doi: 10.1109/JIOT.2020.2974671.

[7] P.-Y. Liu *et al.*, "Associations of PM2.5 and aspergillosis: Ambient fine particulate air pollution and population-based big data linkage analyses," *J. Ambient Intell. Humanized Comput.*, to be published, doi: 10.1007/s12652-018-0852-x.

[8] M. Chen, W. Li, Y. Hao, Y. Qian, and I. Humar, "Edge cognitive computing based smart healthcare system," *Future Gener. Comput. Syst.*, vol. 86, pp. 403–411, Apr. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17326766

[9] L. Greco, P. Ritrovato, and F. Xhafa, "An edge-stream computing infrastructure for real-time analysis of wearable sensors data," *Future Gener. Comput. Syst.*, vol. 93, pp. 515–528, Apr. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X18314031

[10] C. Li, J. Bai, and J. Tang, "Joint optimization of data placement and scheduling for improving user experience in edge computing," *J. Parallel Distrib. Comput.*, vol. 125, pp. 93–105, May 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0743731518302661

[11] Z. Yin, H. Chen, and F. Hu, "An advanced decision model enabling two-way initiative offloading in edge computing," *Future Gener. Comput. Syst.*, vol. 90, pp. 39–48, Jan. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17329527

[12] X. Li and J. Wan, "Proactive caching for edge computing-enabled industrial mobile wireless networks," *Future Gener. Comput. Syst.*, vol. 89, pp. 89–97, Dec. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17329588

[13] M. A. Zamora-Izquierdo, J. Santa, J. A. Martinez, V. Martinez, and A. F. Skarmeta, "Smart farming IoT platform based on edge and cloud computing," *Biosyst. Eng.*, vol. 177, pp. 4–17, Jan. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1537511018301211

[14] S. Wang, Y. Zhao, J. Xu, J. Yuan, and C.-H. Hsu, "Edge server placement in mobile edge computing," *J. Parallel Distrib. Comput.*, vol. 127, pp. 160–168, May 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0743731518304398

[15] Y. Shao, C. Li, and H. Tang, "A data replica placement strategy for IoT workflows in collaborative edge and cloud environments," *Comput. Netw.*, vol. 148, pp. 46–59, Jan. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128618311460

[16] X. Shi, W. Lu, Y. Zhao, and P. Qin, "Prediction of indoor temperature and relative humidity based on cloud database by using an improved bp neural network in Chongqing," *IEEE Access*, vol. 6, pp. 30559–30566, 2018.

[17] P. P. R. Filho *et al.*, "Energy production predication via Internet of Things based machine learning system," *Future Gener. Comput. Syst.*, vol. 97, pp. 180–193, Aug. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X18328486

[18] C.-T. Yang, J.-C. Liu, S.-T. Chen, and K.-L. Huang, "Virtual machine management system based on the power saving algorithm in cloud," *J. Netw. Comput. Appl.*, vol. 80, pp. 165–180, Feb. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804516302971

[19] C.-T. Yang, S.-T. Chen, J.-C. Liu, R.-H. Liu, and C.-L. Chang, "On construction of an energy monitoring service using big data technology for the smart campus," *Clust. Comput.*, vol. 23, no. 1, pp. 265–288, 2020, doi: 10.1007/s10586-019-02921-5.

[20] C.-T. Yang, S.-T. Chen, W. Den, Y.-T. Wang, and E. Kristiani, "Implementation of an intelligent indoor environmental monitoring and management system in cloud," *Future Gener. Comput. Syst.*, vol. 96, pp. 731–749, Jul. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X18304187

[21] S. Yang, J. Bian, L. Wang, H. Zhu, Y. Fu, and H. Xiong, "EdgeSense: Edge-mediated spatial-temporal crowdsensing," *IEEE Access*, vol. 7, pp. 95122–95131, 2018.
[22] S. A. Hossain, M. A. Rahman, and M. A. Hossain, "Edge computing framework for enabling situation awareness in IoT based smart city," *J. Parallel Distrib. Comput.*, vol. 122, pp. 226–237, Dec. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0743731518306130
[23] M. Babazadeh, "Edge analytics for anomaly detection in water networks by an Arduino101-LoRa based WSN," *ISA Trans.*, vol. 92, pp. 273–285, Sep. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0019057819300278
[24] D. G. Roy, B. Mahato, D. De, and R. Buyya, "Application-aware end-to-end delay and message loss estimation in Internet of Things (IoT)—MQTT-SN protocols," *Future Gener. Comput. Syst.*, vol. 89, pp. 300–316, Dec. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17329990
[25] M. Nasir, K. Muhammad, J. Lloret, A. K. Sangaiah, and M. Sajjad, "Fog computing enabled cost-effective distributed summarization of surveillance videos for smart cities," *J. Parallel Distrib. Comput.*, vol. 126, pp. 161–170, Apr. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0743731518308402
[26] R. Morabito, R. Petrolo, V. Loscra, and N. Mitton, "Reprint of: LegIoT: A lightweight edge gateway for the Internet of Things," *Future Gener. Comput. Syst.*, vol. 92, pp. 1157–1171, Apr. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X18325123
[27] Y. Song, J. Lin, M. Tang, and S. Dong, "An Internet of Energy things based on wireless LPWAN," *Engineering*, vol. 3, no. 4, pp. 460–466, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2095809917306057
[28] T. Addabbo, A. Fort, M. Mugnaini, E. Panzardi, A. Pozzebon, and V. Vignoli, "A city-scale IoT architecture for monumental structures monitoring," *Measurement*, vol. 131, pp. 349–357, Jan. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0263224118307978
[29] H. Ponce and S. Gutiarrez, "An indoor predicting climate conditions approach using Internet-of-Things and artificial hydrocarbon networks," *Measurement*, vol. 135, pp. 170–179, Mar. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0263224118310972
[30] C.-T. Yang, C.-W. Huang, and S.-T. Chen, "Improvement of workload balancing using parallel loop self-scheduling on Intel Xeon Phi," *J. Supercomput.*, vol. 73, no. 11, pp. 4981–5005, 2017.
[31] M. Tiwary, D. Puthal, K. S. Sahoo, B. Sahoo, and L. T. Yang, "Response time optimization for cloudlets in mobile edge computing," *J. Parallel Distrib. Comput.*, vol. 119, pp. 81–91, Sep. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0743731518302430
[32] E. Kristiani, C.-T. Yang, and C.-Y. Huang, "iSec: An optimized deep learning model for image classification on edge computing," *IEEE Access*, vol. 8, pp. 27267–27276, 2020.

Endah Kristiani (Graduate Student Member, IEEE) received the M.S. degree in electrical engineering (information technology) from Universitas Gadjah Mada, Yogyakarta, Indonesia, in 2007. She is currently pursuing the Ph.D. degree with the Department of Industrial Engineering and Enterprise Information and join the High Performance Computing Laboratory, Tunghai University, Taichung, Taiwan.

In August 2007, she joined the Faculty of Engineering and Computer Science, Department of Informatics Engineering, Krida Wacana Christian University, West Jakarta, Indonesia.

Chao-Tung Yang (Member, IEEE) received the Ph.D. degree in computer science from National Chiao Tung University, Hsinchu, Taiwan, in July 1996.

In August 2001, he joined the Faculty of the Department of Computer Science with Tunghai University, Taichung, Taiwan, where he is a Distinguished Professor of computer science. He is serving in a number of journal editorial boards, including *Future Generation Computer Systems*, the *International Journal of Communication Systems*, the *KSII Transactions on Internet and Information Systems*, the *Journal of Cloud Computing*. He has published more than 300 papers in journals, book sections and conference proceedings. His present research interests are in cloud computing, big data, parallel computing, and deep learning.

Dr. Yang is a member of the IEEE Computer Society and ACM.

Chin-Yin Huang received the Ph.D. degree from Purdue University, West Lafayette, IN, USA, in 1999.

He is a Professor and the Department Chairman of Industrial Engineering and Enterprise Information with Tunghai University, Taichung, Taiwan. His research interests include healthcare management, clinical data analysis, distributed manufacturing systems, manufacturing process optimization, and Industry 4.0. His publications appear in *International Journal of Production Research*, *International Journal of Production Economics*, *Computers in Industry*, *Computers and Industrial Engineering*, *Robotics and Computer-Integrated Manufacturing*, *Epilepsy Research*, *Production Engineering*, and *Engineering Computations*. He also coauthored sections for Handbooks of Industrial Engineering, Handbook of Industrial Robotics, and Handbook of Automation. He has coauthored two books in Industrial Engineering and Management published in Taiwan.

Po-Cheng Ko received the B.S. degree in computer science and the M.S. degree in computer science from Tunghai University, Taichung, Taiwan, in 2017 and 2019, respectively.

Halim Fathoni received the master's degree from Computer Science and Information Enterprise, Asia University, Taichung, Taiwan, in 2009. He is currently pursuing the Ph.D. degree with the Department of Industrial Engineering and Enterprise Information, High-Performance Computing Laboratorium, Tunghai University, Taichung.

He has been working with Politeknik Negeri Lampung, Bandar Lampung, Indonesia, since 2010. He is curious about edge computing and edge intelligent.