

## Implementation of Integration VaaMSN and SEMAR for Wide Coverage Air Quality Monitoring

Yohanes Yohanie Fridelin Panduman\*<sup>1</sup>, Adnan Rachmat Anom Besari<sup>2</sup>,  
Sritrusta Sukaridhoto<sup>3</sup>, Rizqi Putri Nourma Budiarti<sup>4</sup>, Rahardhita Widyatra Sudibyo<sup>5</sup>,  
Funabiki Nobuo<sup>6</sup>

<sup>1,2,3</sup>Department of Informatics and Computer Engineering, Politeknik Elektronika Negeri Surabaya,  
Surabaya, Indonesia

<sup>4</sup>Department of Engineering, Nahdlatul Ulama University of Surabaya, Indonesia

<sup>5,6</sup>Graduate School of Natural Science and Technology, Okayama University, Tsushimanaka, Japan

\*Corresponding author, e-mail: dphoto@pens.ac.id

### Abstract

The current air quality monitoring system cannot cover a large area, not real-time and has not implemented big data analysis technology with high accuracy. The purpose of an integration Mobile Sensor Network and Internet of Things system is to build air quality monitoring system that able to monitor in wide coverage. This system consists of Vehicle as a Mobile Sensors Network (VaaMSN) as edge computing and Smart Environment Monitoring and Analytic in Real-time (SEMAR) cloud computing. VaaMSN is a package of air quality sensor, GPS, 4G Wi-Fi modem and single board computing. SEMAR cloud computing has a time-series database for real-time visualization, Big Data environment and analytics use the Support Vector Machines (SVM) and Decision Tree (DT) algorithm. The output from the system are maps, table, and graph visualization. The evaluation obtained from the experimental results shows that the accuracy of both algorithms reaches more than 90%. However, Mean Square Error (MSE) value of SVM algorithm about 0.03076293, but DT algorithm has 10x smaller MSE value than SVM algorithm.

**Keywords:** air quality monitoring, VaaMSN, real-time analytical, internet of things, SEMAR.

**Copyright © 2018 Universitas Ahmad Dahlan. All rights reserved.**

### 1. Introduction

With the increased probability of environmental pollution accidents which are caused by the chemical industrial construction, development of economy and production activity are increasingly frequent leading to increased air pollution accident in an urban area [1]. Indonesia is the 6th largest greenhouse gas emitter in the world (IEA 2015) where 40% Percentage of Energy Emissions from transportation and 90% of transportation emissions comes from road transportation. In 2010, the population of Jakarta is 9.607.787 and 57.8% of the population suffers from various air pollution-related diseases. That problem makes total health cost reach up to 38.5 trillion IDR/USD54 billion [2]. The various air pollution-related diseases arise because of the absence of an air quality monitoring system that covers in a large area and detail to determine areas with poor air quality. The government uses air monitoring sensors that installed on a fixed position air monitoring station to measure air condition. The system covers only a small area and does not cover the whole city, maintenance and repair are difficult. The system still uses a conventional database not use Big Data environment, and also, they use an operator to maintain and protect the sensor devices. There are 5 air quality parameters in Indonesian air pollution rules, there is carbon monoxide (CO), nitrogen dioxide (NO<sub>2</sub>), sulfur dioxide (SO<sub>2</sub>), ozone (O<sub>3</sub>) and particulate (PM<sub>10</sub>) [3]. These parameters have an impact on human health.

One of the ongoing research on real-time environmental monitoring and analysis with integrated by big data technology called Smart Environment Monitoring and Analytic in Real-time System (SEMAR) [4-6]. SEMAR can be integrated with water quality monitoring system on ROV (a small robot submarine) [4], this research was developed to solve the government problem on river conditions monitoring. Water quality sensor was mounted on ROV to take samples of water quality and send the result to the server. The server of SEMAR used the big data storage to collected and saved data from sensors in the Hadoop server by utilizing

HDFS, Yarn, and Map Reduce [5]. The researchers also built a SEMAR's extension for water quality monitoring in real-time with integrating Internet of Things (IoT) and Big Data analytic [6]. The data result is classified using a machine learning algorithm. In addition to SEMAR research, there are also other studies on the application of the Internet of Things for environmental monitoring, namely research on location-based environmental monitoring using GPS sensors and Big Data Technology to detect coral reef damage [7] and research on wireless sensor network integration with cloud computing for monitoring. Water quality environment that collaborates with smart aeration systems [8].

However, the system is only implemented in water environmental monitoring and is not used for extensive area monitoring. Therefore, the SEMAR's extension can be improved for other environmental monitoring such as air quality condition monitoring. This system can combine with a mobile sensor [9] to improve the area of monitoring. This journal has five sections that organized as follows: Section 1 presents about Intro of the research, section 2 presents the related works and previous study of the research, section 3 presents the detail of our system design that we used in our research, section 4 presents the results and discussion of the experimental and the implementation of our research. And for section 5 we present conclusions including the recommendation of the future work that will be conducted for the extensions of our project research.

## 2. Related Works

The study and previous works about the air quality monitoring smart system that have been developed using various technology depend on the protocol and also using the real-time system. In 2016, researchers from India have conducted a study of implement Internet of Things on Smart Pollution Detection with AWS IOT Cloud, where the air quality sensor and GPS installed on the vehicle [10]. This system used for tracking and detecting air pollutants in urban, they don't use machine learning or any classification.

Mobile Enterprise Sensor Bus (M-ESB) is a research from China which used to urban environment sensing such as road condition and air pollution. M-ESB send the results of the data sensor that installed on a bus to the server and stored into a database, the output generated from this system is a display in the form of electronic maps website [11]. In 2017, Zhihan Lv and team [12] have conducted the study about Big Data Analytics challenges and the future topics of Big Data development. The result of this study shows that trends of IoT-Big Data platform are the retrieval data process that focused more and more on streaming and multiple sensors data. MapReduce and machine learning algorithm used in the analysis method of Big Data. In this research, we also conducted the research by utilized an IoT-Big Data platform.

## 3. Research Method

Figure 1 shows the IoT reference model [13] which consists of seven sections. This model used as a reference for the design of our system.

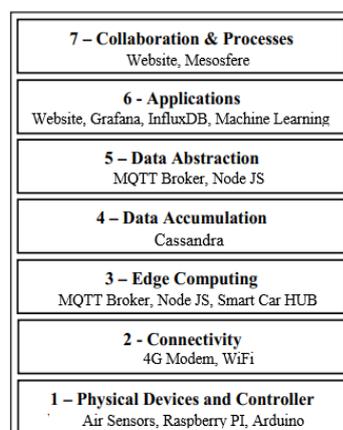


Figure 1. IoT reference model.

Figure 1 describes the layering systems of IoT Reference Model. The system grouped according to the IoT reference model consisting of physical devices and controllers (1), connectivity (2), edge computing (3), data accumulation (4), data abstraction (5), application (6), collaboration and process (7). The overall system design shows in Figure 2. Figure 2 shows the overall system divided into several layers according to the IoT reference model [13] in Figure 1. We will explain briefly about implementation according to layering system in IoT Reference Model in the next sub-section.

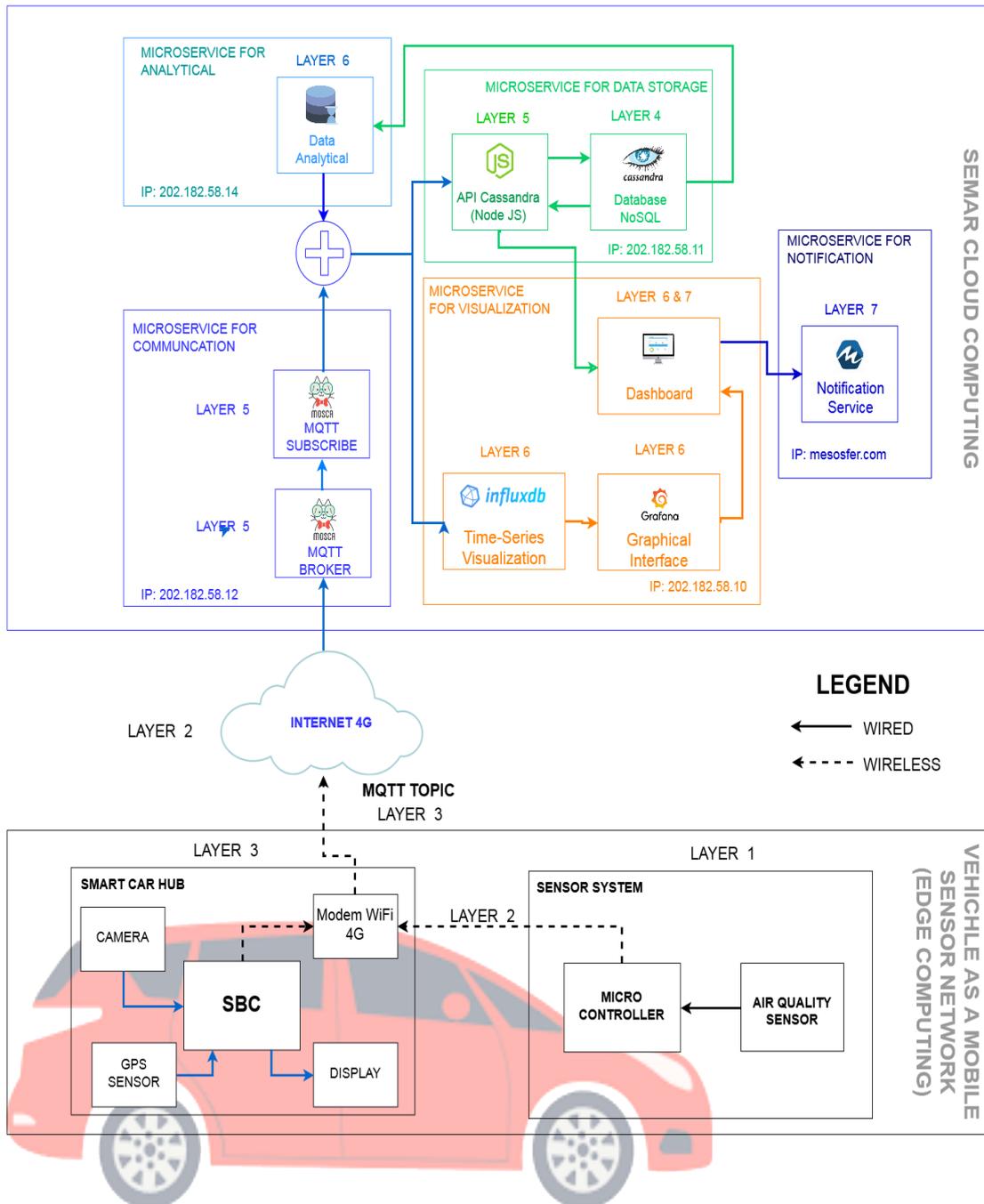


Figure 2. Architecture of integration VaaMSN with SEMAR cloud system

### 3.1. Physical Devices and Controller

Physical Devices and Controllers are part of architecture IoT that used to detect air quality condition. The device has been equipped with the air quality sensors such as Shinyei PPD42 Particulate Matter Detector for particulate, MQ7 for carbon monoxide, MQ135 for sulfur dioxide, MQ131 for ozone and MiCS 2714 for nitrogen dioxide. The device consists of an air quality sensor which is controlled by a microcontroller [14]. The data collection process started with air quality measurement using air quality sensors based on wireless sensor network [15]. Wireless sensor networks are used because they could be implemented in urban areas [16]. This controller collects the data and performs the conversion process for the air quality unit ug/m<sup>3</sup>. The converted data sent to another device for processing to the server via Wi-Fi network. In this research, physical devices and controllers are installed on top of the vehicle and retrieving data from the sensors every 5 seconds.

### 3.2. Connectivity

Through the Wi-Fi signal from a 4G modem, Physical devices and controllers send data to the edge computing layer for processing. The 4G modem is also used to connect between Single Board Computing (SBC) on layers the edge computing and cloud computing. The throughput of this 4G modem is about 20 Mbps.

### 3.3. Edge Computing

Smart car hub consists of SBC, Camera, GPS, display and Wi-Fi 4G modem an Edge Computing [17]. The data collected by the sensor is combined with GPS data in JSON format, then send the data through the MQTT protocol.

#### Algorithm 1. Single Board Computing

```
1: Begin:
2:   MQTT Connect
3:   loop:
4:     Read sensor data
5:     Read GPS Sensor
6:     Add Sensor data and GPS Sensor to JSON Format
7:     Send data to the server using MQTT (topic,
8:     data)
8: End
```

SBC is the main controller to connect VaaMSN system with cloud computing system, where there are other devices connected to it such as Camera that serves to take picture condition around the car, as well as a display screen that serves as a user interface. In this research, Raspberry Pi 3 type B is used as SBC, Raspberry Pi sends a line of air quality data through the topic 'airsensor' to MQTT broker on cloud computing. The data lines transmitted has formats like 'air quality data, latitude, longitude and current time'.

### 3.4. Data Accumulation

Data accumulation is a process of storing data which sent by edge computing to the No SQL database as a Big data platform. Air quality data in JSON form, received by MQTT Subscriber with topic 'airsensor'. The data is used for the prediction process to determine the air quality index in accordance with the rules in Section 2.8. Air Quality Index results obtained from predicted results are stored in Cassandra that serves as No SQL database with schemas consist of a current timestamp, air quality data, latitude, longitude, label.

Cassandra was chosen because it has stable performance, strong security, operational simplicity for the lowest total cost of ownership and best scalability of NoSQL platforms [18]. The data with topic 'airsensor' stored on Cassandra through an Application Programming Interface (API) web service and the use of the Representative State Transfer (REST) communication architecture method that runs on the Node.js service in microservice for data storage with IP address 202.182.58.11.

### 3.5. Data Abstraction

This layer defines the management of the data flow for Big Data server and real-time visualization. We built two connector applications the first connector is based on Node.js that

subscribed into MQTT Broker [19]. This connector function is to get data from MQTT Broker, distributed data to Cassandra database server by using RESTful API and send data to Influx DB visualization server as backend. The second connector is based on Python programming language. This connector function subscribes the data from MQTT Broker to Machine Learning server, and send the predicted data to MQTT Broker with a different topic. The IP addresses of microservices are 202.182.58.10 for InfluxDB as real-time visualization, 202.182.58.12 for MQTT Broker and connector, 202.182.58.14 for RESTful API and Cassandra database server.

### 3.6. Applications

The application layer consists of 1) Learning Process, 2) Real-time classification, 3) Real-time visualization in the form of a table, map, and graph.

#### 3.6.1. Learning Process

The process of Machine Learning is used to build the classification model. This process is used to provide a classification model before performing the classification process in real-time. Thus, the level of confidence in classification results can affect the accuracy level of the generated model.

In the training process of the dataset, we used Scikit-learn [20] for conducting the training process of the dataset. Support Vector Machine [21-23] and Decision Tree [24-26] are used as classification algorithms for this research. The best method between the two would be selected by comparing the results of the algorithm.

Support Vector Machine (SVM) used for regression and classification algorithm [20], this algorithm has been implemented for big data classification [23]. In SVM, this classification is performed by giving a training vectors  $y_i \in R^n$ ,  $i = 1, \dots, l$ , with an indicator vector  $y_i \in R^l$  such that  $y_i \in \{1, -1\}$  to solves the following primal optimization problem:

$$\begin{aligned} \min_{w, b, \xi} \quad & \frac{1}{2} w^T w + C \sum_{i=1}^n \xi_i \\ \text{subject to: } & y_i (w^T \phi(x_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, i = 1, \dots, l, \end{aligned} \quad (1)$$

Due to the high dimensional possibilities of the vector variables  $w$ , where  $C > 0$  is the parameter of regularization and  $\phi(x_i)$  map  $x_i$  to the higher dimension. we can solve the following double problem:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \alpha^T Q \alpha + e^T \alpha \\ \text{subject to } & y^T \alpha = 0 \\ & 0 \leq \alpha_i \leq C. \quad i = 1, \dots, l, \end{aligned} \quad (2)$$

after problem (2) is solved, using the primal-dual relationship, the optimal  $w$  satisfies:

$$w = \sum_{i=1}^l y_i \alpha_i \phi(x_i) \quad (3)$$

The decision function of the classification becomes:

$$f(w^T \phi(x) + b) = \text{sign} \left( \sum_{i=1}^l y_i \alpha_i K(x_i, x) + b \right) \quad (4)$$

The original SVM can be classified into two classes. Proper multiclass methods are required when dealing with more than two-class classification problems. In this case, combine several binary classifiers with two methods. The first method is 'One on one' means applying a comparison of inter-class pairs. The second method is 'One against the other' means comparing one class with all the other classes.

The decision tree is a machine learning algorithm that uses tree decisions such as trees and the possible subsequent impact, which involves the results of events, resource costs, and utilities. Decision Tree is one of the best classifiers when considering classification accuracy,

this algorithm studies the classification function which includes the dependent attribute (variable) given by the value of the independent attribute (input) (variable).

Some of the most well-known decision tree algorithms are C4.5, CART and Naive Bayes Tree [24]. This research uses a CART that stands for Classification and Regression Trees. CART analysis is a form of binary recursive partitioning and can handle numerical and categorical variables [24-26]. The impurity level of accepted data can be measured by CART, also it can construct a binary tree where each internal node produces two classes for the accepted attribute. The way of how the tree constructed by selecting the attribute recursively use an attribute which has the lowest Gini Index. Attribute with the lowest Gini Index value is obtained by calculating the Gini Index value in each attribute. Gini Index is calculated based on the formula below, where the probability of the  $i^{th}$  class for  $c$  target classes of a given attribute is  $P_i$ , meanwhile,  $P_i$  is the probability of class  $i$  [24].

$$Gini = 1 - \sum_{i=1}^c (P_i)^2 \quad (5)$$

The accuracy of the classifier algorithm was evaluated by divides the dataset into two subsets are about 70% for the training set and the remaining 30% for the test set. The training set is used to build the classification model. While the measurement of the built classification model performance used test set. The method used is called the hold-out method. Learning procedure is shown in Algorithm 2.

#### Algorithm 2. Single Board Computing

```

1: Method → Linear SVM, Decision Tree
2: Begin:
3:   sensorIndo → Air Quality Sensor
4:   loop:
5:     Retrieve Dataset
6:     Machine Learning Training (method)
7:     Machine Learning Testing (method)
8:     Calculate MSE, Mislabeled, Accuracy
9:     Save model
10: End

```

This research uses Air Pollution Standard Index Range (ISPU), this rule is used by the current Republic of Indonesia government to determine the quality category [3] as shown in Table 1. Parameters of the Air Pollution Standards Index include carbon monoxide (CO), nitrogen dioxide (NO<sub>2</sub>), particulate (PM<sub>10</sub>), ozone (O<sub>3</sub>), and sulfur dioxide (SO<sub>2</sub>). The 'Range' column in Table 1 refers to the measured air quality index formulation values in determining the formulation be explained in Table 2.

Table 1. Air Pollution Standard Index

Category	Range
Good	0-50
Moderate	51-100
Not Healthy	101-199
Very Not Healthy	200-299
Hazardous	300-more

Table 2 describes the groupings of values of each air quality parameter in ug/m<sup>3</sup> units to be formulated to determine the air quality index. The 'Air Quality Index' column shows the maximum value of the air quality index with the air quality parameter conditions in accordance with the air quality parameter columns in the same row.

Determining the air quality index by the value grouping rules and value constraints can be used in (6) as set out in Table 2.

$$I = \frac{I_a - I_b}{X_a - X_b} (X_x - X_b) + I_b \quad (6)$$

Explanation:

- I : Air pollution index are counted.
- Ia : Air pollution index upper limit.
- Ib : lower ground air pollution index Ambient upper limit.
- Xa: Ambient upper bound.
- Xb: Ambient lower bound.
- Xx : Real ambient content of measurement results.

Table 2. Air Quality Index

Air Quality Index	PM <sub>10</sub> ug/m3	SO <sub>2</sub> ug/m3	CO ug/m3	O <sub>3</sub> ug/m3	NO <sub>2</sub> ug/m3
50	50	80	5	120	-
100	150	365	10	235	-
200	350	800	17	400	1130
300	420	1600	34	800	2260
400	500	2100	36	1000	3000
500	600	2620	57,5	1200	3750

In determining the ISPU, if there are several air quality parameters that are measured then, the data used is the air quality parameter with the highest ISPU value. For example, if the data obtained SO<sub>2</sub> = 71, NO<sub>2</sub> = 55, PM<sub>10</sub> = 91 then the reported data is ISPU worth 91, Air Quality is "Moderate" and the dominant parameter is PM<sub>10</sub>.

### 3.6.2 Real-time Classification

Through the data model that has been generated by the learning process and analytics on large-scale data can be used to create a real-time classification system. Therefore, although data is used on a large scale with a large number of sensor nodes, the system able still perform the analysis process. The purpose of using this system is to bypass the data distribution delay from VaaMSN (edge computing) to data storage and visualization.

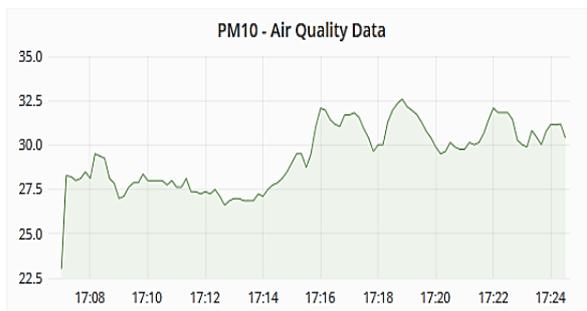
The process of real-time classification and learning process using a scikit-learn that runs in the python environment. Air quality data sent by VaaMSN on the topic of 'airsensor' through MQTT communication, then the data is converted into JSON format so that it can be used in classification to generate air quality index prediction from received data. The results of the process are numerical from 0 to 4 representing the categories in Table 1, sequentially starting from good, medium, bad, very bad and hazardous. The result is stored in a variable called 'label and put into JSON previously received data so that the data contains 'air quality data, latitude, longitude and the current time and labels'. The combined data be re-submitted with the topic "airsensoranalytic" for use in real-time visualization.

### 3.6.3 Real-Time Visualization

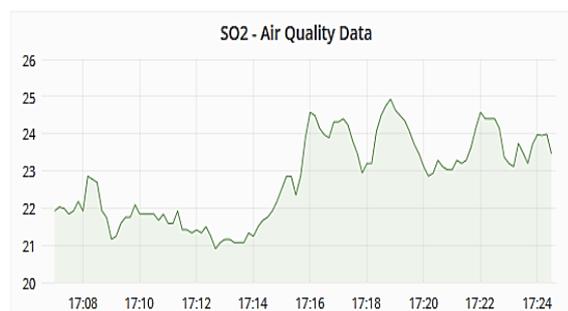
Visualization stage start from the connector sends the data into InfluxDB by using 'writepoint()' function on Node.js. We use InfluxDB for time-series database [27], the data collected by InfluxDB as arranged by time-series, then send to Grafana [28]. The Grafana generated graphical interfaces such as a table, graph, and maps. Data schemes are "{current timestamp, sensor id, pm10, co, so2, o3, no, latitude, longitude, latitude, label}". We built three type of visualization, Figure 3 (a) is a table, show data of air monitoring sensors with latitude, longitude and index quality air monitoring, Figure 3 (b-f) is a Graph show the time series data with a line chart, and Figure 3 (g) is a Map show point of VaaMSN.

Air Quality Table					
Time	sensor.CO	sensor.SO2	sensor.Latitue	sensor.Longitude	sensor.Kondisi
2018-01-23 20:27:00	10.54	17.07	-7.31	112.75	4
2018-01-23 20:26:50	10.24	16.66	-7.32	112.75	3
2018-01-23 20:26:40	10.24	16.66	-7.32	112.75	3
2018-01-23 20:26:30	10.09	16.45	-7.32	112.75	3
2018-01-23 20:26:20	9.95	16.25	-7.32	112.75	2
2018-01-23 20:26:10	9.67	15.84	-7.32	112.75	2
2018-01-23 20:26:00	9.67	15.84	-7.32	112.75	2
2018-01-23 20:25:50	9.67	15.84	-7.32	112.75	2
2018-01-23 20:25:40	9.67	15.84	-7.32	112.75	2

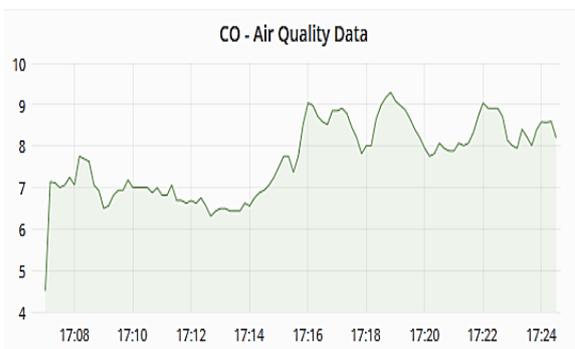
(a)



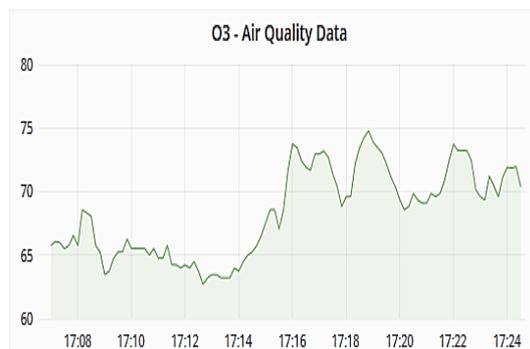
(b)



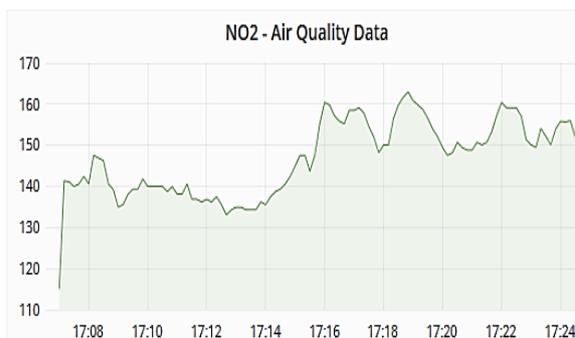
(c)



(d)



(e)



(f)



(g)

Figure 3. Visualization: (a) table (b-f) graph and (g) map.

### 3.7. Collaboration and Processes

This layer provides feedback to users performed by the system. The system will send a notification if there is bad air quality in an Air Quality Parameter area. If there is a label value exceeding bad characteristics of the data sent on the 'airqualityanalytic' topic, the website visualization system built able to provide a notification to the user in the form of a warning that the current air condition is hazardous or not healthy. Not only using the website to push notification but also can be using Mesosfer platform. The Mesosfer is Mobile Backend Platform as a Service that provides several features to help and simplify the creation of the internet of things system and enabling users to speed up the development process. This platform used to send mobile notifications to alert the user about air quality condition. When air quality is in unhealthy, very unhealthy and hazardous condition system would send data to Mesosphere RestAPI, then the data becomes notification sent by Mesosphere to the user's cell. The data submitted is 'air quality data', 'air quality condition', 'current time' and 'data location'.

## 4. Results and Discussion

In this results sections, we have done some experiment and present the implementation both of software and hardware development. The experiment has been given the results that performed as well and using the analytical test showed how the system that we built works well. Several tests performed including real-time visualization testing and comparative performance evaluation of SVM algorithms [29] and linear Decision Tree using their default parameters where we use datasets from sensors according to the given rules.

### 4.1. Results of Confusion Matrix

Table 3 shows the confusion matrix of the result of the training model that has been built from the Linear SVM algorithm, Table 4 shows the confusion matrix for the DT algorithm. The SVM experiment results that the number of data error are between 22 until 127 from around 21.804 data. It is mean that the error percentage is about 0.2% to 0.5%. The confusion matrix of SVM can be seen in Table 3.

Table 3. Confusion Matrix of Support Vector Machine Dataset Training.

		Predicted Class				
		Good	Moderate	Not Healthy	Very Unhealthy	Hazardous
Actual Class	Good	4716	22	0	0	0
	Moderate	22	4110	0	0	0
	Not Healthy	11	100	4129	0	0
	Very Unhealthy	0	11	127	3932	18
	Hazardous	0	0	3	85	4518

Table 4. Confusion Matrix of Decision Tree Dataset Training.

		Predicted Class				
		Good	Moderate	Not Healthy	Very Unhealthy	Hazardous
Actual Class	Good	4732	6	0	0	0
	Moderate	3	4331	1	0	0
	Not Healthy	0	12	4225	3	0
	Very Unhealthy	0	0	3	4079	6
	Hazardous	0	0	0	5	4601

The Decision Tree algorithm results in experimental results that the number of data error is between 3 until 12 from around 21.804 data. It is mean that the error percentage is almost 0%. Table 4 showed that the confusion matrix of the Decision Tree algorithm. The experiment showed that Decision tree is better than SVM with higher accuracy of the predicted label.

### 4.2. Classification Results

The second experiment we measure the acceleration of the classification result. To measure the acceleration we calculate the accuracy rate and MSE (Mean Squared Error). Table 5 shows that Decision tree algorithms offer a better accuracy rate by 0.99839479 when

compared to the accuracy level of the SVM algorithm that around 0.98170061. However, according to Table 5, both algorithms show good performance with an accuracy rate over 90% and MSE is 0.00268096 for decision tree algorithms and 0.03076293 for SVM algorithm.

Figure 4 shows the curve shape of the ROC (Receiver Operating Curve) which represents the Validation of the class model. ROC is a curve that compares graphs on the vertical axis of TPR (True Positive Rate) with FPR (False Positive Rate) existing on the horizontal axis of ROC. The area under the ROC curve called AUC. AUC is rated from 0 to 1 and gets better when it approaches 1. From the experiments performed, the decision tree algorithm has 100% accuracy in all classes, while the SVM algorithm has an accuracy of around 98%. These AUC results are better compared to the use of multilabel classifier [30] which produces AUC of around 0.71.

Table 5. Comparison of Two Algorithms on the Dataset.

Features	Algorithm	Mislabel	Accuracy	MSE
Air Quality	Support Vector Machine	399 / 21804	0.98170061	0.03076293
	Decision Tree	35 / 21804	0.99839479	0.00268096

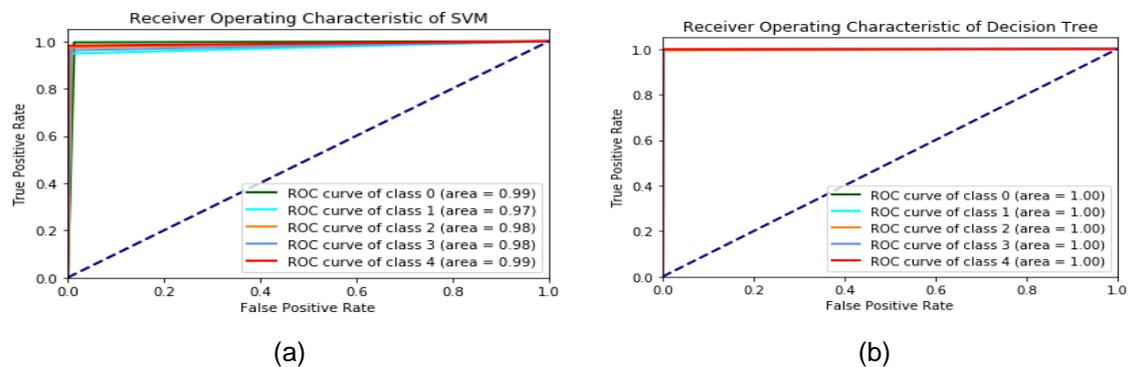


Figure 4. Receiver Operating Characteristics: (a) Using SVM, and (b) Using Decision Tree.

From Figure 4 (a) is showed ROC graphical results of SVM algorithm SVM accuracy value of each class about 97% to 99%. Meanwhile, Figure 4 (b) is show the ROC graphics results from the decision tree algorithm, obtained from the image of the accuracy value of each class is about 100%.

### 4.3. Purpose System Implementation

Experiments are used to test system integration from sensor readings to database storage and real-time visualization. Air quality data obtained by air quality sensor devices is that transmitted through communication to cloud computing. MQTT Broker distributes data to the prediction system to determine the air quality index of the data. The data that has been sent back after the data added with air quality index use another topic to MQTT Broker and accepted by MQTT Customer and forwarded to InfluxDB connector for real-time visualization. Data received by InfluxDB displayed in real-time using graphical interface graphics that can be viewed on the website at IP address 202.182.58.10.

The test is performed by installing an air quality sensor device on top of the vehicle and SBC in the vehicle. Vehicles are driven on the highway for collecting air quality data in the road. Figure 5 (a) shows the air quality sensor device and Figure 5 (b) shows the SBC as Smart Car Hub. Figure 6 shows real-time visualization on dashboard when the vehicle is in motion, a maker's visualization on the world map representing air quality data, the color of the marker shows the air quality data with the provision is green for good condition, blue for medium condition, yellow for unhealthy, red for very no healthy and black for hazardous.

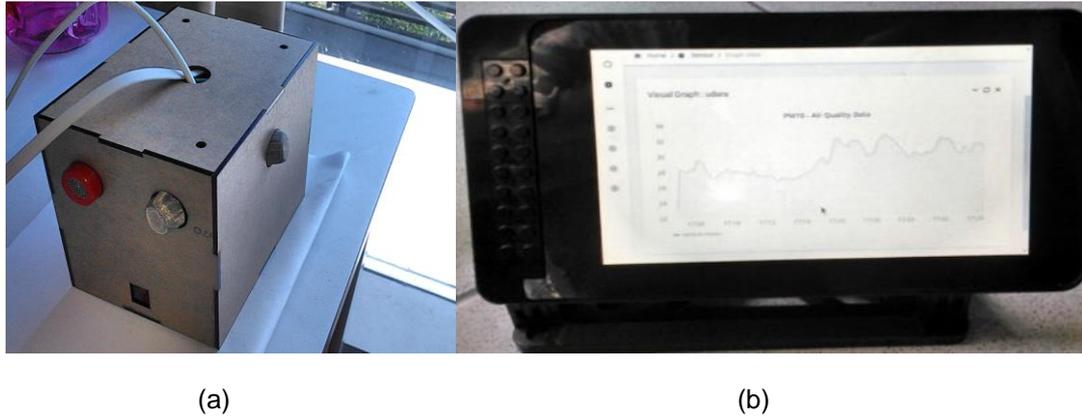


Figure 5. Devices: (a) air quality sensor, and (b) smart car Hub

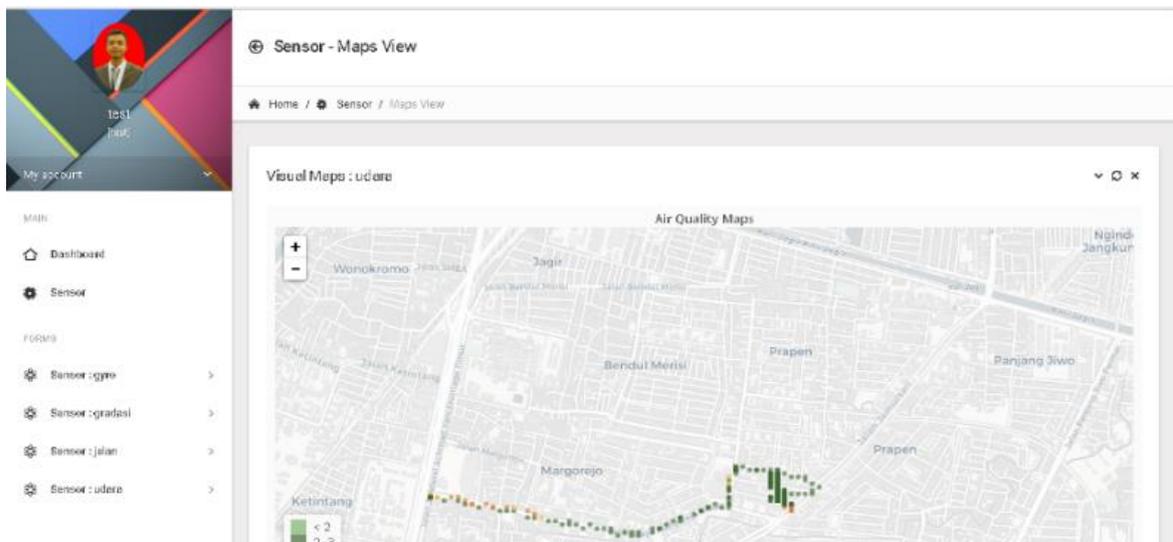


Figure 6. Dashboard.

## 5. Conclusion

In this research, the integration of VaaMSN and SEMAR for the air quality monitoring system has been implemented successfully. From the experiment show that the data from VaaMSN is sent to Big Data platform and visualized in real-time. The two algorithms that we used for analytical have been given the result that the estimation of accurate is more than 90% and achieves MSE is 0.00268096 for DT algorithm and SVM algorithm about 0.03076293. That means we achieve a good result in this experiment. In the future, the integration of VaaMSN and SEMAR is expected to be used in the road environmental monitoring which detects holes and road damage using cameras and other road condition sensors in real-time classification.

## 6. Acknowledgment

This research is funded by KEMERISTEKDIKTI from "Applied Leading Research in Higher Education (in Indonesia Penelitian Terapan Unggulan Perguruan Tinggi)" in 2018 scheme with the number: 09/PL14/PG.1/SP2P/2018 and the title is "Integrated as a Mobile Sensor Network Vehicle Implementation with Smart Environment Monitoring and Analytics in Real-time (SEMAR) system as Road Surface Monitoring and Environment to support Smart City (in Indonesia Implementation Vehicle as a Mobile Sensor Network terintegrasi dengan Smart Environment Monitoring and Analytics in Real-time (SEMAR) system sebagai Pemantauan

Permukaan Jalan dan Lingkungan untuk mendukung Smart City)”. This research is also funded by DITJEN PENGUATAN INOVASI from "Innovation Strengthening Program (in Indonesia Program Penguatan Inovasi)" in 2018 scheme with the number: 08/PPK/SK/INOVASI INDUSTRI-DII/III/2018 with the title is "Development of Fin-lot (Financial Technology and Internet of Things) Platforms as Supporters of Digital-Based Payments for Small Medium Enterprise (in Indonesia Pengembangan Platform Fin-lot (Financial Technology Dan Internet of Things) Sebagai Pendukung Pembayaran Berbasis Digital Untuk Small Medium Enterprise))"

## References

- [1] Chen Xiaojun, Liu Xianpeng, Xu Peng. *IOT-based Air Pollution Monitoring and Forecasting System*. International Conference on Computer and Computational Sciences. Nanjing. 2015: 257–260.
- [2] Satya Widya Yudha. *Air Pollution in Indonesia Challenges and Imperatives for Change*. Singapore: The National Bureau of Asian Research.2016.
- [3] Head of Environmental Impact Control Agency, *Calculation and Reporting As well as Information on Air Pollution Standard Index (in Indonesia Perhitungan Dan Pelaporan Serta Informasi Indeks Standard Pencemar Udara)*. Indonesia: of Environmental Impact Control Agency.1997.
- [4] Sukaridhoto S, Pramadihanto D, Alif M, Yuwono A, Funabiki N. *A design of radio-controlled submarine modification for river water quality monitoring*. Intelligent Technology and Its Applications (ISITIA). Surabaya. 2015: 75-80.
- [5] Berlian MH, Sahputra TE, Ardi BJ, Dzatmika LW, Besari AR, Sudibyo RW, Sukaridhoto S. *Design and implementation of smart environment monitoring and analytics in real-time system framework based on internet of underwater things and big data*. in electronics Symposium (IES). Bali. 2016: 403-408.
- [6] Riyadh Arridha, Sritrusta Sukaridhoto, Dadet Pramadihanto, Nobuo Funabiki. *Classification Extension based on IoT-Big Data Analytic for Smart Environment Monitoring and Analytic in Realtime System*. *Int. J. Space-Based Situated Comput*. 2017; 7(2): 82-97.
- [7] Abid Famasya Abdillah, et al. *Design and Development of Low Cost Coral Monitoring System for Shallow Water Based on Internet of Underwater Things*. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*. 2017; 9.2-5: 97-101.
- [8] Dzulqornain, Muhammad Iskandar, M Udin Harun Al Rasyid, Sritrusta Sukaridhoto. *Design and Development of Smart Aquaculture System Based on IFTTT Model and Cloud Integration*. In: MATEC Web of Conferences. EDP Sciences, 2018 :01030.
- [9] H L Brantley, G S W Hagler, E S Kimbrough, R W Williams, S Mukerjee, L M. Neas. *Mobile air monitoring data-processing strategies and effects on spatial air pollution trends*. *Atmospheric Measurement Techniques*. 2014 ; 7(1): 2169-2183.
- [10] Sruthi MM, Mary LJ. *Smart pollution detection and tracking system embedded with AWS IOT cloud*. *International Journal of Advanced Research in Computer Science and Software Engineering*. 2016; 6(4): 1-20.
- [11] Kang L, Poslad S, Wang W, Li X, Zhang Y, Wang C. *A public transport bus as a flexible mobile smart environment sensing platform for IoT*. Intelligent Environments (IE). Londo. 2016; 1-8.
- [12] Lv Z, Song H, Basanta-Val P, Steed A, Jo M. *Next-generation big data analytics: State of the art, challenges, and future research topics*. *IEEE Transactions on Industrial Informatics*. 2017; 13(4): 1891-9.
- [13] Jim Green. *Building the Internet of Things | An Overview of the IoT Reference Model*. Chicago: *Internet of Things World Forum*.2014.
- [14] Bochen L, Xibo D, Qingyao C. *Development of air quality monitoring system in closed environment*. *TELKOMNIKA Telecommunication, Computing, Electronics and Control*. 2017; 15(4): 2003-2012.
- [15] Iqbal M, Fuad M, Sukoco H, Alatas H. *Hybrid Tree-Like Mesh Topology as New Wireless Sensor Network Platform*. *TELKOMNIKA Telecommunication, Computing, Electronics and Control*. 2016; 14(3): 1166-1174.
- [16] Bushra Rashid and Mubashir Husain Rehman. *Applications of wireless sensor networks for urban areas: A survey*. *Journal of Network and Computer Applications*, 2016; 60: 192-219.
- [17] Weisong Shi, et al. *Edge computing: Vision and challenges*. *IEEE Internet of Things Journal*, 2016; 3.5: 637-646.
- [18] Rodrigo Aniceto, et al. *Evaluating the Cassandra NoSQL database approach for genomic data persistency*. *International journal of genomics*, 2015; 2015.
- [19] Hunkeler U, Truong HL, Stanford-Clark A. *MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks*. *Communication systems software and middleware and workshops*. Bangalore. 2008: 791-798.

- 
- [20] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J. Scikit-learn: Machine learning in Python. *Journal of machine learning research*. 2011; 12: 2825-2830.
- [21] Corinna Cortes, Vladimir Vapnik. Support-vector networks. *Machine learning*. 1995; 20(3): 273-297.
- [22] Chang CC, Lin H. A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*. New York. 2011; 3(2):27(1-27).
- [23] A. Suárez Sánchez, P.J. García Nietob, P. Riesgo Fernández, J.J. del Coz Díaz, F.J. Iglesias-Rodríguez. Application of an SVM-based regression model to the air quality study at local scale in the Avilés urban area (Spain). *Mathematical and Computer Modelling*. 2011; 54(5-6): 1453-1466.
- [24] B Saba Bashir, Usman Qamar, Farhan Hassan Khan, M.Younus Javed. *An Efficient Rule-Based Classification of Diabetes Using ID3, C4. 5, & CART Ensembles*. 12th International Conference of Frontiers of Information Technology (FIT). Islamab. 2014: 226-231.
- [25] David H. Hagan, Gabriel Isaacman-VanWertz, Jonathan P. Franklin, Lisa M. M. Wallace, Benjamin D. Kocar, Colette L. Heald, Jesse H. Kroll. Calibration and assessment of electrochemical air quality sensors by co-location with reference-grade instruments. *Atmospheric Measurement Techniques*. 2018; 11(1): 315-328.
- [26] D. Birant. Comparison of Decision Tree Algorithms for Predicting Potential Air Pollutant Emissions with Data Mining Models. *Journal of Environmental Informatics*, 2011; 17(1):46-53.
- [27] Benjamin Leighton, Simon Cox, Nicholas J. Car, Matthew P. Stenson, Jamie Vleeshouwer and Jonathan Hodge. *A best of both worlds approach to complex, efficient, time series data delivery*. In: International Symposium on Environmental Software Systems. Springer. 2015: 371-379.
- [28] Gennaro Tortone, et al. *Web-Based Visualization Tools for Epics Embedded Systems: An Application to Belle2*. In: 16th Int. Conf. on Accelerator and Large Experimental Control Systems (ICALEPCS'17), Barcelona, Spain, 8-13 October 2017. JACOW, Geneva, Switzerland, 2018. p. 1857-1860.
- [29] Akash Saxena and Shalini Shekhawat. Ambient air quality classification by grey wolf optimizer based support vector machine. *Journal of environmental and public health*, 2017; 2017: 3131083.
- [30] Giorgio Corani and Mauro Scanagatta. Air pollution prediction via multi-label classification. *Environmental Modelling & Software*, 2016; 80: 259-264.