Classification Extension based on IoT-Big Data Analytic for Smart Environment Monitoring and Analytic in Real-time System

Riyadh Arridha*

Department of Information and Computer Engineering
Politeknik Elektronika Negeri Surabaya
Surabaya, Indonesia 60111
E-mail: riyadh@pasca.student.pens.ac.id
*Corresponding author

Sritrusta Sukaridhoto

Department of Multimedia Creative Technology
Politeknik Elektronika Negeri Surabaya
Surabaya, Indonesia 60111
E-mail: dhoto@pens.ac.id

Dadet Pramadihanto

Department of Information and Computer Engineering
Politeknik Elektronika Negeri Surabaya
Surabaya, Indonesia 60111
E-mail: dadet@pens.ac.id

Nobuo Funabiki

Department of Electrical and Communication Engineering
Okayama University
Okayama, Japan 700-8530
E-mail: funabiki@okayama-u.ac.jp

Abstract: Monitoring water conditions in real-time is a critical mission to preserve the water ecosystem in maritime and archipelagic countries, such as Indonesia that is relying on the wealth of water resources. To integrate the water monitoring system into the big data technology for real-time analysis, we have engaged in the ongoing project named SEMAR (Smart Environment Monitoring and Analytic in Real-time system), which provides the IoT-Big Data platform for water monitoring. However, SEMAR does not have an analytical system yet. This paper proposes the analytical system for water quality classification using Pollution Index method, which is an extension of SEMAR. Besides, the communication protocol is updated from REST to MQTT. Furthermore, the real-time user interface is implemented for visualisation. The evaluations confirmed that the data analytic function adopting the linear SVM and Decision Tree algorithms achieves more than 90% for the estimation accuracy with 0.019075 for the MSE.

Keywords: SEMAR; water condition monitoring; real-time analysis; IoT; big data; classification; machine learning.

Biographical notes: Riyadh Arridha received the Bachelor's degree in information engineering from Universitas Islam Negeri Alauddin Makassar, Indonesia, in 2011. He joined at Politeknik Negeri Fakfak, Indonesia, as a lecturer in 2013. He is now studying for his Master's degree in information and computer engineering at Politeknik Elektronika Negeri Surabaya, Indonesia. His research interests include big data, Internet of Things and embedded system.

Sritrusta Sukaridhoto received the B.E. degree in electrical engineering, computer science program from Sepuluh Nopember Institute of Technology, Indonesia, in 2002 and the Ph.D. degree in Communication Networks Engineering from Okayama University, Japan, in 2013. He joined at Politeknik Elektronika Negeri Surabaya, Indonesia, as a lecturer in 2002, and He became an Assistant Professor in 2011, respectively. He stayed at Tohoku University, Japan, in 2004, as a visiting researcher. His research interests include computer networks, embedded system, multimedia and Internet of Things. He has received several academic awards, best paper awards

and IEEE Young Researcher Award in 2009. He is a member of IEEE.

Dadet Pramadihanto received his M.Eng. and Ph.D. degrees from the Osaka University, Japan in 1997 and 2003, respectively. From 1998 to 1999, he was also researcher at the Kansai Laboratory of Image Information Systems and Technology, Japan and worked on the project of Face Recognition Systems. From 2003, he was Associate Professor at the Department of Information and Computer Engineering, Politeknik Elektronika Negeri Surabaya, Indonesia. From 2005 to 2009 he served as Vice Director on Academic Affairs and from 2009 to 2013 he served as Director of the Politeknik Elektronika Negeri Surabaya. Now, he holds the Head of Robotics Research Center at the Politeknik Elektronika Negeri Surabaya.

Nobuo Funabiki received the B.S. and Ph.D. degrees in mathematical engineering and information physics from the University of Tokyo, Japan, in 1984 and 1993, respectively. He received the M.S. degree in electrical engineering from Case Western Reserve University, USA, in 1991. From 1984 to 1994, he was with Sumitomo Metal Industries, Ltd., Japan. In 1994, he joined the Department of Information and Computer Sciences at Osaka University, Japan, as an assistant professor, and became an associate professor in 1995. He stayed at University of Illinois, Urbana-Champaign, in 1998, and at University of California, Santa Barbara, in 2000–2001, as a visiting researcher. In 2001, he moved to the Department of Communication Network Engineering (currently, Department of Electrical and Communication Engineering) at Okayama University as a professor. His research interests include computer networks, optimisation algorithms, educational technology, and web technology. He is a member of IEEE and IPSJ.

1 Introduction

Monitoring water conditions in real-time is a critical mission to preserve the water ecosystem in maritime and archipelagic developing countries, including Indonesia that is relying on the wealth of water resources. For example, in Indonesia, approximately 70% of the overall area is water, and a soaring rainfall is expected every year due to the crossing of the equator line. The potential wealth of water resources is crucial to support Indonesian's life sustainability.

At the same time, Indonesia is also facing serious problem of the lack of awareness in preserving water resources (Hapsari et al., 2016). The environmental conditions of water have become increasingly critical. The clean water crisis is actually extended from year to year. In the rainy season, excessive water causes flooding, meanwhile in the dry season, water sources dry up. As time goes by, water resources may no longer be available due to pollutions.

To solve the above-mentioned problem, we have engaged in the ongoing project named *SEMAR (Smart Environment Monitoring and Analytic in Real-time system)*. SEMAR is a real-time system based on IoT (Internet of Things) and big data for monitoring and analysing water conditions. SEMAR can be used by the parties included in the decision making. SEMAR consists of water quality monitoring system using ROV (Remotely Operated Vehicle), wireless mesh network, portable water quality monitoring system, coral reef monitoring system, and big data storage system.

However, SEMAR does not have an analytical system yet. In this paper, we propose the big data analytic system as SEMAR's extension in real-time. Besides, we updated SEMAR's transmission protocol from REST (Representational State Transfer) to MQTT (Message Queuing Telemetry Transport) (Banks and Gupta, 2014). Furthermore, we built the user interface which has an ability to handle the real-time data.

This paper is organised as follows: Section II reviews some related works in literatures. Section III introduces our works of SEMAR. Section IV explains the proposed system. Section V shows experiment results for evaluations. Section VI concludes this paper with future works.

2 Related Works

Modaresi and Araghinejad (2014) have conducted the study of the water quality classification using CCME (Canadian Council of Minister of the Environment) Water Quality Index, with two parameters: nitrate and chloride. In this study they used three algorithms: SVM (Support Vector Machine), Probabilistic Neural Network, and K-Nearest Neighbour. The result revealed that SVM shows the best performance without any error in calibration and validation process.

Ladjal et al., (2016) have also conducted the study of water quality classification using Dempster-Shafer theory. In this study, Ladjal et al., used four parameters: temperature, pH, conductivity, and turbidity, and used Neural Network and SVM. The result of this study indicated that SVM has better performance than Neural Network.

Jaloree et al., (2014) have conducted another study on water quality classification using decision tree algorithm. As the parameters in determining the water quality, they used pH, DO, BOD, No3_N, and NH3_N. The results of the training process showed 95.4545% accuracy rate.

Saghebian et al. (2014) have conducted a study of groundwater quality classification by using decision tree algorithm, based on the USSL (United States Salinity Laboratory) diagram. The results showed that the overall average of CCI (Correctly Classified Instances) and Kappa Statistic for prediction of the groundwater quality classes based on the USSL diagram were 0.88 and 0.83 %, respectively. Unfortunately, all of the four studies above did

not support real-time classification and did not integrate with big data technology.

Fazio et al., (2015) have conducted the study of implementing big data as the storage for smart environment monitoring system. This study, presented in general, was about the sensor integrated system in the cloud environment for Advanced Multi-risk Management (SIGMA) which is a part of the Italian National Operative Program (PON). This project was expected to accommodate all sorts of data from various environment. Unfortunately, there is no clear descriptions of the implementation of the big data analytic.

Moore et al., (2016) have conducted a study on real-time monitoring and big data solutions for storage and predictions. They implemented IAL (Independent Assisted Living) and patient monitoring system.

Richter et al., (2015) have conducted several comparative studies on toolkits for machine learning in big data, including Mahout MapReduce, Mahout Samsara, Spark MLlib, H2O, and SAMOA. The study showed that on average, Spark MLlib and H2O has better performances than other toolkits in terms of the extensibility, scalability, usability, fault tolerance and speed. Spark MLlib implements 17 algorithms, Mahout MapReduce does 13, H2O does 10, Mahout Samsara does 7, and SAMOA does 3. Another advantage of Spark MLlib is the ability to cover the batch and stream processing. Conversely, Mahout and H2O only cover the batch processing, and SAMOA only covers the stream processing.

Our system does not use conventional web and database services. Instead, big data technology is adopted, where HDFS is used as the file system for fast access. Besides, other technologies to support real-time processing and integrated with machine learning technology are incorporated. The big data technology allows diverse, large, and fast data to be addressed in more than one computer.

3 SEMAR

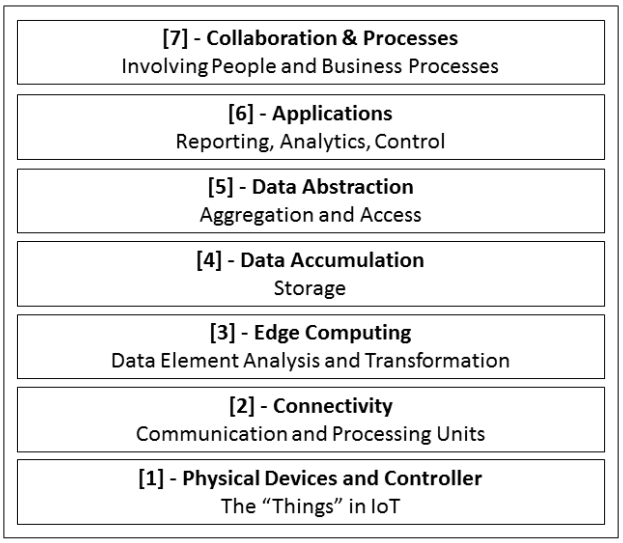
In SEMAR, water quality monitoring system using ROV (a small robot submarine) (Sukaridhoto et al., 2015) was developed to solve the problem in the government in taking samples to monitor the river conditions. The ROV with water quality sensors can be controlled remotely. The operator did not have to take samples manually. The sampling results from sensors can be delivered directly to the server by using internet.

The wireless mesh network (Yuliandoko et al., 2016) was also adopted to extend the range of data communication between sensors and the server. The portable water quality monitoring system (Sukaridhoto et al., 2016) was developed as a portable low-cost COTS-based system that able to send data to the server directly. The coral reef monitoring system (Abdillah et al., 2016) was used to monitor the condition of coral in shallow water by using an active camera and directly send data to the server. The big data storage server (Berlian et al., 2016) collected and saved data from sensors in Hadoop server by utilising HDFS, Yarn and Map Reduce.

3.1 System Overview

Figure 1 shows the IoT reference model (Cisco, 2014). Our system design is based on the IoT reference model which consists of seven sections: 1) physical devices and controllers, 2) connectivity, 3) edge computing, 4) data accumulation, 5) data abstraction, 6) application and 7) collaboration and processes. Next, Figure 2 shows the overall system design. Physically, the infrastructure of this system consists of one master node and two slave nodes.

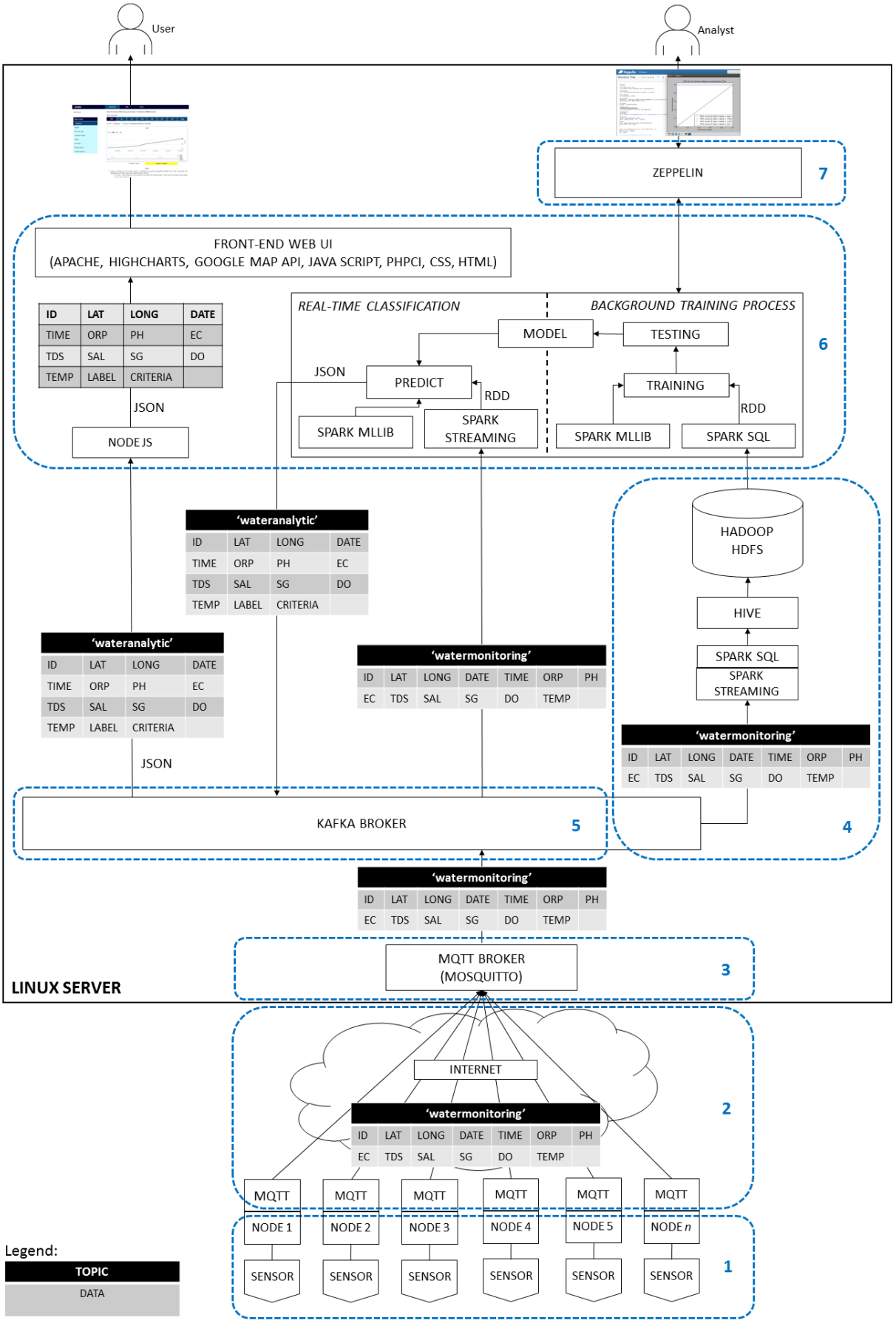
Figure 1 IoT reference model



3.1.1 Physical Devices and Controllers

Physical Devices and Controllers are the nodes that have been equipped with water quality sensors scattered at several points along the river that runs through the city of Surabaya. The water quality sensors come from 'Atlas Scientific' (Atlas, 2017) kit water sensor. The 'Atlas Scientific' kit includes pH (Potential of Hydrogen), ORP (Oxidation Reduction Potential), DO (Dissolved Oxygen), EC (Electrical Conductivity), and Temperature. The nodes use Raspberry Pi 3, type B. In this research, there are six point nodes scattered along the river and retrieving data from the sensors in every 5 seconds. The arrangement of data sent uses comma as a separator with these orders: Sensor ID, Latitude, Longitude, Date, Time, ORP, pH, EC, TDS, Sal, SG, DO, Temperature. The data communication protocol uses MQTT.

Figure 2 System Design.



The sensor nodes will act as MQTT publishers which send the data to MQTT broker (server). The communication port uses 1883, and the topic of data sent is 'watermonitoring'. The algorithm of Physic Sensor App can be seen on algorithm 1.

Algorithm 1 Physic Sensor App Procedure.

```
1:  Begin
2:  sensorInfo ← sensorID, latitude, longitude,
3:  loop:
4:      time ← system time
5:      sensorData ← pH,DO,EC,ORP,Temp,TDS,Sal,Sg
6:      Send sensorInfo, time, sensorData to MQTT Server
7:      Save local
8:      Sleep (5)
9:      Goto loop.
10: End
```

3.1.2 Connectivity

The 4G modem is used to connect between the embedded system and the server. The throughput of this 4G modem is around 20 Mbps.

3.1.3 Edge Computing

The process of data acceptance in the server uses MQTT broker by Mosquitto (Light, 2013), which is sent by the sensor nodes. The water data is received and stored in the 'watermonitoring' topic.

3.1.4 Data Accumulation

Data Accumulation is a process of storing water data from sensor nodes to the Hadoop HDFS (Hadoop, 2009). Spark Streaming (Spark, 2015) consumes/subscribes water data from the Kafka Broker (Kafka, 2014) on topic 'watermonitoring'. The received water data by Spark Streaming is buffered in 10 seconds. Every 10 seconds, the data is loaded into Hadoop HDFS using Hive (Hive, 2017). The querying of Hive is executed using Spark SQL.

In Hadoop HDFS, the data is saved into the table named 'watermonitoringku'. Ten seconds are allocated to give more time to the system for Map Reduce processing. Hive is used for data saving because it uses SQL queries. Besides, in this system, the real-time access is not necessary for data storage. The water's data in Hadoop HDFS which required for further analysis is loaded in batch processing using Hive Query. The water data is consumed/subscribed from Kafka Broker, so there is a mechanism for distributing data from MQTT Broker (Mosquitto) to Kafka Broker directly. MQTTKafkaBridge (Kalaria, 2016) is used to bridge the distribution of topic 'watermonitoring' from MQTT to Kafka Broker.

3.1.5 Data Abstraction

Kafka Broker is used to managing the data flow on the big data server. Thus, the water data from sensor nodes with the topic 'watermonitoring' in MQTT Broker (Mosquitto) is distributed to the Kafka Broker beforehand. The direct distribution of water data with the topic 'watermonitoring' from MQTT Broker to Kafka Broker is applied using MQTTKafkaBridge. Then, Kafka Broker produces/ publishes the data to the consumer/subscriber applications. Kafka Broker also uses the same scheme as MQTT but with different terminology. Kafka Producer stands for data sender, Kafka Consumer stands for those who take data and Kafka Broker stands as an intermediary. The data is stored in certain topics. The topic for data coming from sensor nodes is 'watermonitoring', while the topic for data analysis result is 'wateranalytic'. The water data is not retrieved from data storage, with the intention to speed up the flow of data so the real-time processing can be realised.

3.1.6 Application

The application layer of SEMAR consists of; 1) Learning Process, 2) Real-time classification, and 3) Real-time visualisation. Previously, SEMAR used the conventional visualisation which could not handle the real-time data. In this current research, we upgrade the visualisation of SEMAR to the real-time visualisation.

3.1.7 Collaboration and Processes

At this stage, there are some interactions between the analyst and the system. As an interface of this section, it uses Zeppelin (Zeppelin, 2017). Zeppelin is an open-source web-based notebook to analyse and visualise in an interactive way. By integrating Zeppelin with Apache Spark (Sharma, 2016), the building of the classification model is performed through the web in an interactive way.

3.2 Open Source Software

In SEMAR system, we utilise open source software in the sensor, communication, big data server, analytic system, visualisation and for the application development. Mostly, we use the Python library to build SEMAR system.

4 Proposed System

In this section, we explain our proposed system on the Application layer. Our proposed system consists of; 1) Learning Process, which describes the building of the classification model used in the system, 2) Real-time classification, which describes the schema of real-time analysis and 3) Real-time visualisation, which describes the real-time visualisation process on the front-end web user interface.

4.1 Learning Process

The learning process is the stage for building the classification model which used in the system. This process is done before the real-time classification processing. Then, the accuracy level of the generated model greatly affects the level of confidence in classification results.

This paper uses data which retrieved from PDAM Surya Sembada Surabaya (PDAM, 2016). The data are laboratory test data and live sensor data.

The data of laboratory test is the data from the daily laboratory test result which conducted in 2014 to 2016 with 1,347 samples and 20 attributes. The laboratory test data consists of: date, temperature, turbidity, colour, SS, pH, alkalinity, CO2 free, DO, nitrite, ammonia, copper, phosphate, sulphide, iron, hexavalent chromium, manganese, zinc, lead, and COD.

The data of live sensor is the data from sensors which placed in several places in Surabaya's river, the data is taken on March to August in 2016 with 205.720 samples and 6 attributes. The live sensor data consists of: date-time, turbidity, TSS, pH, DO, and temperature.

The determination of the class label uses Pollution Index method. Pollution Index is one of the two methods recognised in Indonesia in determining water quality. It can be calculated by eq. (1) (Hidup, 2003).

$$PI_j = \sqrt{\frac{\left(C_i/L_{ij}\right)_M^2 + \left(C_i/L_{ij}\right)_R^2}{2}} \quad (1)$$

L_{ij} is the concentration of water quality parameter in water designation standard (j), and C_i is expressed as the concentration of water quality parameter (i) obtained from the measurement result of a river channel. PI_j is Pollution Index for designation (j), it is the function of $\frac{C_i}{L_{ij}}$ and determined from the resultant maximum value (M) and the mean value (R) concentration ratio per parameter of the value of the water quality standard. Evaluate of the value PI_j can determine the categories of Pollution Index as shown in eq. (1). The category can be seen in Table 1.

Table 1 Pollution Index Category.

No.	Pollution Index	Categories
1	$0 \leq PI_j \leq 1.0$	Fulfil Standard
2	$1.0 < PI_j \leq 5.0$	Lightly Polluted
3	$5.0 < PI_j \leq 10.0$	Polluted
4	$PI_j > 10.0$	Heavy Polluted

In laboratory test and live sensor data, parameters used in determining Pollution Index are turbidity, TSS, pH, DO, and temperature. These five parameters are standard parameters used by PDAM Surya Sembada Surabaya in monitoring the condition of raw water before being processed into drinkable water. From the Pollution Index category, there are four possible labels, namely; Fulfil Standard, Lightly Polluted, Polluted, and Heavy Polluted. Then, the label is translated into the numerical to facilitate the training process as Fulfil Standard = 0, Lightly Polluted = 1, Polluted = 2, and Heavy Polluted = 3.

In the training process of the dataset, Spark MLlib is used for conducting training process of the dataset. By using Spark SQL, data analysis could be performed on large scale. Spark MLlib has supported several algorithms that could be used for classification, clustering, or regression. As proposed by (Modaresi and Araghinejad, 2014), (Ladjal et al., 2016), (Jaloree et al., 2014), (Saghebian et al., 2014), we use two classification algorithms, where the results would be compared to choose the best one. Classification algorithms which would be used were Support Vector Machine and Decision Tree.

Support Vector Network (Cortes and Vapnik, 1995) or Support Vector Machine (SVM) is has been extensively used for classification and regression (Suykens and Vandewalle, 1999). On the binary SVM, the classification is performed by the optimal linear separating hyperplane between two classes. If (x_i, y_i) is dataset with $i = 1, \dots, n$, where x_i is the vector containing m features, and $y_i \in \{-1,1\}$ is the class label related to x_i. SVM solves the following primal problem (Modaresi and Araghinejad, 2014), (Ladjal et al., 2016), (Suykens and Vandewalle, 1999).

$$Minimise \frac{1}{2}|w|^2 + c\sum_{I=1}^{n}\xi_i$$
$$Subject\_to: y_i(wx + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i \in \{1, \dots, n\} \quad (2)$$

The problem is changed to the following dual problems by using the Lagrange multipliers (Modaresi and Araghinejad, 2014), (Suykens and Vandewalle, 1999).

$$Maximise \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i=1.j=1}^{n} a_i a_j y_i y_j x_i x_j$$
$$Subject\_to: \sum_{i=1}^{n} a_i y_i = 0, 0 \leq a_i \leq C, \forall i \in \{1, \dots, n\} \quad (3)$$

The level of error in classification is adjusted by C parameter.

The nonlinear transformation $\emptyset$ is done through the kernel function $K(x_i, x_j)$. It describes nonlinearity mapping from the input space to the higher dimensional space features. The dual problem of SVM Lagrange turns into (Modaresi and Araghinejad, 2014), (Ladjal et al., 2016), (Jaloree et al., 2014), (Ito and Kunisch, 2008).

$$Maximise \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i=1.j=1}^{n} a_i a_j y_i y_j K(x_i x_j)$$
$$Subject\_to: \sum_{i=1}^{n} a_i y_i = 0, 0 \leq a_i \leq C, \forall i \in \{1, \dots, n\} \quad (4)$$

After obtaining an optimal parameter a, the decision function of the classification becomes (Modaresi and Araghinejad, 2014), (Ladjal et al., 2016).

$$f(x_j) = sign\left(\sum_{i=1}^{n} a_i y_i K(x_i x_j) + b\right) \quad (5)$$

SVM with the linear kernel is implemented in this research. The formula of Linear Kernel of SVM (Modaresi and Araghinejad, 2014), (Fan et al., 2008) is formulated as follows:

$$k(x_i x_j) = x_i^T x_j.$$

The original SVM can be classified into two classes. If it deals with more than two classes classification problems, an appropriate multiclass method is needed. In this case, combining several binary classifiers with two methods (Modaresi and Araghinejad, 2014), (Hsu and Lin, 2002);

a. 'One against one' means implementing inter-class pair comparisons.
b. 'One against the others' means comparing one class with all the other classes.

One against one method is selected to be used in this study.

The decision tree is a decision support system which uses a tree-like graph decision and its after-effect possibility, involving chance event results, resource costs, and utility. The decision tree or classification tree is used in learning the classification function which infers the dependent attribute (variable) value given by the independent attribute (input) (variable) value.

Some of the well-known decision tree algorithms are ID3, C4.5, SPRINT, SLIQ, C5.0 and CART (Anyanwu and Shiva, 2009). This research uses CART, which is known as the classification and regression trees (Breiman et al., 1984) algorithm, to classify water quality. Both numerical and categorical variables can be handled by CART. It can measure the impurity level of accepted data and construct a binary tree where each internal node produces two classes for the accepted attribute. The Gini index is calculated for each attribute, the attribute with the lowest Gini index is selected as a breaker attribute (Bramer, 2007). Selecting the attribute recursively with the lowest Gini Index is the way of how the tree is constructed. 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 Pi, meanwhile, Pi is the probability of class i (Bashir et al., 2014).

$$Gini\ Index = 1 - \sum_{i=1}^{c} P_i^2 \quad (6)$$

The hold-out method is used for evaluating the classifier accuracy. This method divides the entire dataset into two parts, namely; training set and test set. The training set is a subset of the dataset used to construct the classification model, and the test set is a subset used to measure the performance of the built classification model. Actually, 70% of the dataset is used for the training set and 30% is used for the test set. The learning procedure is shown in algorithm 2.

Algorithm 2 Learning Procedure.

```
method ← Linear SVM, Decision Tree
Begin
    Retrieve Dataset
    Split Dataset (70,30)
    Machine Learning Training (method)
    Machine Learning Testing (method)
    Calculate MSE, Mislabel, Accuracy
    Save Model
End
```

4.2 Real-time Classification

The real-time classification uses big data analytic technology. Therefore, the number of nodes can be increased up to hundreds or even thousands and can perform data analysis on large scale. This process requires the high speed in order to be a stand-alone application with the purpose to cut the delay between the data is being received until it is visualised.

Real-time classification process uses Spark MLlib as well as in the learning process. The generated classification model on 4.1 is loaded before the classification of new data. The water data is read by Spark from Kafka Broker by topic 'watermonitoring' in streaming way, then conducting a classification which produces the prediction of Pollution Index category. The classification result is saved in a variable called 'label'. This label has a numerical type and it is retranslated into the categorical data (Fulfil Standard, Lightly Polluted, Polluted, or Heavy Polluted) stored in the new variable, called 'criteria'. The result of this classification is loaded into Kafka Broker using Kafka Producer on the port 9092 with the topic 'wateranalytic'. The 'wateranalytic' topic is used for real-time visualisation data. The sent data uses JSON format with the arrangement of Sensor ID, Latitude, Longitude, Date, Time, ORP, pH, EC, TDS, Sal, SG, DO, Temperature, Label, Criteria. The algorithm of the real-time classification can be seen on algorithm 3.

Algorithm 3 Real-time Classification.

```
function parsing(data)
    parse ← param1, param2, …, param n
    return parse
end function
function realtimeClassification(parse)
    classify ← Model.predict
    return classify
end function
function sendToKafka(data,classify)
    Send data, classify to Kafka Broker
end function
Begin
    Spark initialization
    Load Model
    data ← pH,DO,EC,ORP,Temp,TDS,Sal,Sg from Kafka Broker
    parsing (data)
    realtimeClassification (parse)
    SendToKafka(data,classify)
End
```

4.3 Visualisation

Further, Node JS (Node, 2017) takes data from the real-time classification result of Kafka Broker using Kafka Consumer with the topic 'wateranalytic'. By using Node JS, the flow of real-time data can be managed to the front end. The Web socket is used in Node JS to send the data from the backend to the front end. In the front end, the standard technology combination is used to create the attractive appearance, like Apache, PHPCI, JavaScript, CSS and HTML. Specifically, Google Map (Map, 2017) API is used to visualise the location of the sensor node and Highcharts (Highsoft, 2015) is used to visualise water data charts. Highcharts supports real-time visualisation for real-time data. In the visualisation, it also displays information about Pollution Index of the water data. The data flow of visualisation can be seen in Figure 3. Figure 4 shows the visualisation of the node location which exists along the river in Surabaya. The central point of the map is (-7.348195, 112.681339). Figure 5 shows the chart of the water data sensor and information of Pollution Index of real-time classification process result.

Figure 3 Data Flow in Visualisation.

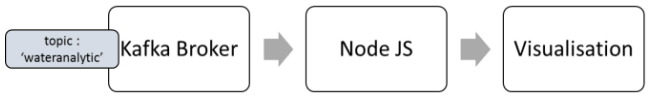


Figure 4 Visualisation of Node Location.

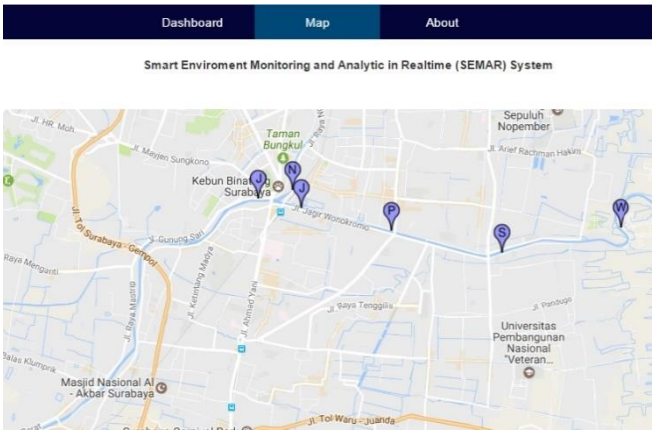
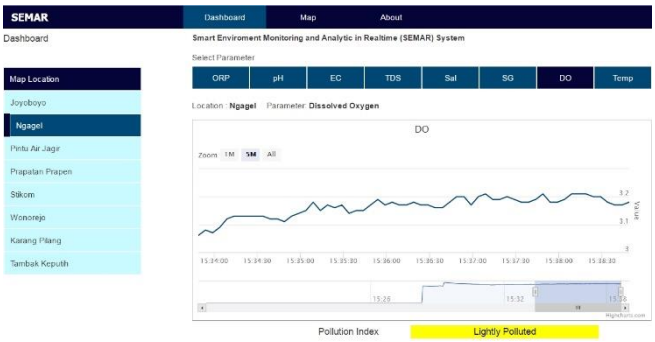


Figure 5 Visualisation of Water Data Sensor and Information of Pollution Index.



5 Experiments and Results

The experiments were conducted by testing the performance of Linear SVM and the Decision Tree algorithm using their default parameters. We used the dataset from PDAM Surya Sembada Surabaya (The laboratory test dataset and the live sensor dataset).

5.1 Confusion Matrix Results

Tables 2 and 3 show the confusion matrix for the laboratory test dataset by Linear SVM and by the Decision Tree algorithm respectively. Tables 4 and 5 show the confusion matrix of the live sensor dataset by Linear SVM and Decision Tree algorithm respectively. The laboratory test data and live sensor data do not meet the fulfil standard (class 0) and the heavy polluted standard (class 4).

Table 2 Confusion Matrix of laboratory test dataset uses Linear Support Vector Machine.

		Predicted Class			
		0	1	2	3
Actual Class	0	0	0	0	0
	1	0	206	6	0
	2	0	19	159	0
	3	0	0	0	0

Table 3 Confusion Matrix on laboratory test dataset uses Decision Tree.

		Predicted Class			
		0	1	2	3
Actual Class	0	0	0	0	0
	1	0	212	0	0
	2	0	2	176	0
	3	0	0	0	0

Table 4 Confusion Matrix of live sensor dataset uses Linear Support Vector Machine.

		Predicted Class			
		0	1	2	3
Actual Class	0	0	0	0	0
	1	0	19770	107	0
	2	0	182	25338	0
	3	0	0	0	0

Table 5 Confusion Matrix of live sensor dataset uses Decision Tree.

		Predicted Class			
		0	1	2	3
Actual Class	0	0	0	0	0
	1	0	19859	18	0
	2	0	16	25504	0
	3	0	0	0	0

5.2 Classification Results

In the two datasets, the labelling results only produced two water classes, lightly polluted (class 1) and polluted (class 2). This means that the river water in Surabaya is at either class 1 or class 2.

Table 6 shows the number of mislabeled data, the accuracy rate, and the MSE (Mean Squared Error) by each algorithm on each dataset that were calculated from the confusion matrix.

Both algorithms show the good performance with the accuracy rate of more than 90% and the MSE around 0.019075. If the results are compared, the decision tree algorithm offers the better accuracy rate of 0.999251 for the live sensor dataset and 0.994872 for the laboratory test dataset.

Table 6 Comparison of two algorithms on laboratory test and live sensor dataset.

Features	Dataset	Algorithm	Mislabel	Accuracy	MSE
pH, TSS, DO, Temp, Turbidity	Laboratory test	Linear Support Vector Machine	25 / 390	0.935897	0.0641
		Decision Tree	2 / 390	0.994872	0.0051
	Live Sensor	Linear Support Vector Machine	289 / 45397	0.993634	0.0064
		Decision Tree	34 / 45397	0.999251	0.0007

Figure 6 ROC of laboratory test dataset using Linear Support Vector Machine and Decision Tree.

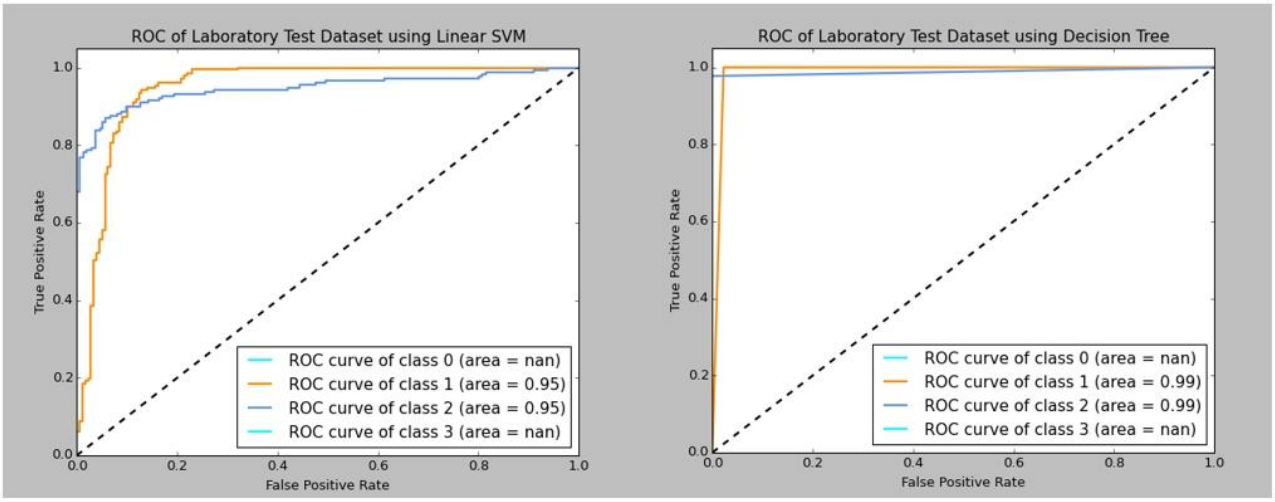
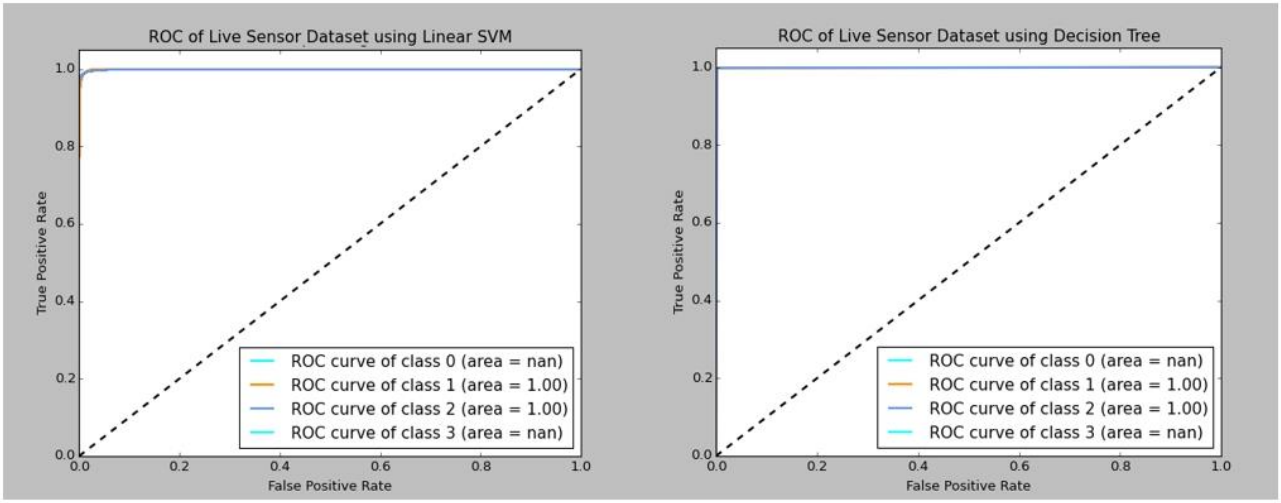


Figure 7 ROC of live sensor dataset using Linear Support Vector Machine and Decision Tree.



The validation of classification model can also be done with ROC (Receiver Operating Curve). It is a comparison graph between TPR (True Positive Rate) on the vertical axis and FPR (False Positive Rate) on the horizontal axis of the ROC. The area under the ROC curve is known as the AUC (Area Under the ROC Curve). The AUC value ranges from 0 to 1. The closer to 1 means the better test value in the classification model. Figure 6 shows the ROC for the laboratory test dataset by both algorithms. Figure 7 shows ROC for the live sensor dataset by them.

By the decision tree algorithm, the ROC graph of live sensor dataset shows 1.00 for class 1 and class 2, and for the laboratory test dataset shows 0.99 for class 1 and class 2. It

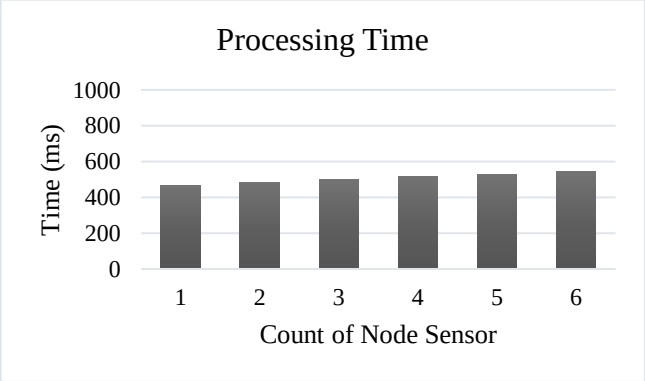
means that the water quality classification system using the decision tree algorithm has the excellent performance with $0.9 \leq AUC \leq 1$.

Spark is a big data machine learning toolkit that uses in-memory processing schema to perform data analysis. The problem that sometimes happens is the memory allocation of Spark's job is less than the required for data analysis. This can lead to Out of Memory Error. It is necessary to increase the Java Heap Space. In this research, Spark is used in a yarn-client mode. The Yarn is used to manage the job of spark, including the memory allocation. At Yarn, 4 GB memory is allocated for each job and we did not meet any problem in building the classification model.

5.3 Processing Time

The server only take approximately 508 milliseconds for all nodes for data processing at the visualisation stage because of the direct data flow to real-time processing. They come from the Spark's ability to process the data by in-memory processing scheme, and the Node JS's ability to support the real-time data flow for visualisation. The processing time did not include the transmission time from the node to the server, which is about 1 second. Thus, it is concluded that the system of the monitoring and real-time classification is effective. Figure 8 shows the average processing time on the server after the water sensor data is received by the server from the node until it ends at the time the data is visualised.

Figure 8 Server Processing Time.



6 Conclusions and Future Work

In this paper, the classification extension based on IoT-Big data analytic for smart environment monitoring and analytic in real-time system has been conducted by integrating IoT and big data. The evaluations confirmed that the data analytic function adopting the linear SVM and the decision tree algorithms achieves more than 90% for the estimation accuracy with 0.019075 for the MSE. In future, SEMAR is expected to be used in the air environment, and for real-time clustering in mapping the water conditions in the river.

7 Acknowledgment

The authors would like to thanks to ER2C (EEPIS Robotics Research Center) for the preparation and source code modification. This research is funded by KEMRISTEKDIKTI from Foreign Cooperation in 2017 scheme with the number: 01/PL14/PG.1/SP2P/2017.

8 References

Abdillah, Abid, Muhammad Herwindra, Yohanes Panduman, Muhammad Akbar, Marlanisa Afifah, Sritrusta Sukaridhoto, Shiori Sasaki. "Design and Development Low-Cost Coral Monitoring System for Shallow Water Based on Internet of Underwater Things." In *Advanced Research in Electronic Engineering and Information Technology International Conference (AVAREIT)*, 2016

Anyanwu, Matthew N., and Sajjan G. Shiva. "Comparative analysis of serial decision tree classification algorithms." International Journal of Computer Science and Security 3, no. 3 (2009): 230-240.

Atlas (2017) *Atlas Scientific* [Online] http://www.atlasscientific.com, (accessed 17 April 2017).

Banks, Andrew, and Rahul Gupta. "MQTT Version 3.1. 1." OASIS standard (2014).

Bashir, Saba, Usman Qamar, Farhan Hassan Khan, and M. Younus Javed. "An Efficient Rule-Based Classification of Diabetes Using ID3, C4. 5, & CART Ensembles." In Frontiers of Information Technology (FIT), 2014 12th International Conference on, pp. 226-231. IEEE, 2014.

Berlian, Muhammad Herwindra, Tegar Esa Rindang Sahputra, Buyung Jofi Wahana Ardi, Luhung Wahya Dzatmika, Adnan Rachmat Anom Besari, Rahardhita Widyatra Sudibyo, and Sritrusta Sukaridhoto. "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), 2016 International*, pp. 403-408. IEEE, 2016

Bramer, Max. Principles of data mining. Vol. 180. London.: Springer, 2007.

Breiman, L., Friedman, J., Stone, C.J. and Olshen, R.A. (1984) *Classification and Regression Trees*, CRC Press, Florida, United States.

Cisco, The Internet of Things Reference Model. White Paper (2014)

Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." Machine learning 20, no. 3 (1995): 273-297.

Fan, R-E., Chang, K-W., Hsieh, C-J., Wang, X-R. and Lin, C-J., 'LIBLINEAR: a library for large linear classification', *Journal of Machine Learning Research*, Vol. 9 No. Aug, pp.1871–1874.

Fazio, M., Celesti, A., Puliafito, A. and Villari, M. (2015) 'Big data storage in the cloud for smart environment monitoring', *Procedia Computer Science*, Vol. 52, pp.500–506.

Hadoop (2017) *Apache Hadoop* [online] http://hadoop.apache.org. (accessed 6 March 2017).

Hapsari, Ratih Indri, and Mohammad Zenurianto. "View of flood disaster management in Indonesia and the key solutions." American Journal of Engineering Research 5, no. 3 (2016): 140-151.

Hidup, Kementerian Negara Lingkungan. "Keputusan Menteri Negara Lingkungan Hidup Nomor 115 Tahun 2003 Tentang Pedoman Penentuan Status Mutu Air." Jakarta: Kementerian Negara Lingkungan Hidup (2003).

Highsoft, A.S. (2017) *Highcharts* [online] http://www.highcharts.com/products/highcharts, (accessed 10 April 2017).

Hive (2017) *Apache Hive* [online] http://hive.apache.org. (accessed 10 March 2017).

Hsu, Chih-Wei, and Chih-Jen Lin. "A comparison of methods for multiclass support vector machines." IEEE transactions on Neural Networks 13, no. 2 (2002): 415-425.

Ito, K. and Kunisch, K. (2008) *Lagrange multiplier approach to variational problems and applications*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, United States.

Jaloree, Shailesh, Anil Rajput, and Sanjeev Gour. "Decision tree approach to build a model for water quality." Binary Journal of Data Mining & Networking 4, no. 1 (2014): 25-28.

Kafka (2017) *Apache Kafka: A Distributed Streaming Platform* [online] http://kafka.apache.org, (accessed 10 March 2017)

Kalaria, D. (2017) *MQTTKafkaBridge* [Online] http://github.com/, DhruvKalaria/MQTTKafkaBridge/ (accessed 13 April 2017).

Ladjal, Mohamed, Mohamed Bouamar, Mohamed Djerioui, and Youcef Brik. "Performance evaluation of ANN and SVM multiclass models for intelligent water quality classification using Dempster-Shafer Theory." In Electrical and Information Technologies (ICEIT), 2016 International Conference on, pp. 191-196. IEEE, 2016.

Light, R. (2017) *Mosquitto-an open source mqtt v3.1/v3.1.1 broker* [online] http://mosquitto.org. (accessed 17 April 2017)

Map (2017) *Google Maps for every platform* [online] http://developers.google.com/maps/ (accessed 10 April 2017).

Modaresi, Fereshteh, and Shahab Araghinejad. "A comparative assessment of support vector machines, probabilistic neural networks, and K-nearest neighbor algorithms for water quality classification." *Water resources management 28*, no. 12 (2014): 4095-4111.

Moore, Philip, Andrew Thomas, George Tadros, Fatos Xhafa, and Leonard Barolli. "Detection of the onset of agitation in patients with dementia: real-time monitoring and the application of big-data solutions." International Journal of Space-Based and Situated Computing 3, no. 3 (2013): 136-154.

Node (2017) *Node JS* [online] http://nodejs.org/ (accessed 10 April 2017).

PDAM (2017) *PDAM Surya Sembada Surabaya* [online] http://www.pdam-sby.go.id/ (accessed 22 March 2017).

Richter, Aaron N., Taghi M. Khoshgoftaar, Sara Landset, and Tawfiq Hasanin. "A multi-dimensional comparison of toolkits for machine learning with big data." In *Information Reuse and Integration (IRI), 2015 IEEE International Conference on*, pp. 1-8. IEEE, 2015.

Saghebian, S. Mehdi, M. Taghi Sattari, Rasoul Mirabbasi, and Mahesh Pal. "Ground water quality classification by decision tree method in Ardebil region, Iran." Arabian Journal of Geosciences 7, no. 11 (2014): 4767-4777.

Sharma, M. Abhishek, and Monica O. Joshi. "Openstack Ceilometer Data Analytics & Predictions." In Cloud Computing in Emerging Markets (CCEM), 2016 IEEE International Conference on, pp. 182-183. IEEE, 2016.

Spark (2017) *Apache Spark: Lightning-fast Cluster Computing* [online] http://spark.apache.org. (accessed 25 March 2017)

Sukaridhoto, Sritrusta, Dadet Pramadihanto, Muhammad Alif, Andrie Yuwono, and Nobuo Funabiki. "A design of radio-controlled submarine modification for river water quality monitoring." *In Intelligent Technology and Its Applications (ISITIA), 2015 International Seminar on*, pp. 75-80. IEEE, 2015.

Sukaridhoto, Sritrusta, Rahardhita Widyatra Sudibyo, Widi Sarinastiti, Rizky Dharmawan, Atit Sasono, Ahmad Andika Saputra, and Shiori Sasaki. "Design and development of a portable low-cost COTS-based water quality monitoring system." In *Intelligent Technology and Its Applications (ISITIA), 2016 International Seminar on*, pp. 635-640. IEEE, 2016.

Suykens, Johan AK, and Joos Vandewalle. "Least squares support vector machine classifiers." Neural processing letters 9, no. 3 (1999): 293-300.

Yuliandoko, H., Sukaridhoto, S., Al Rasyid, M.U.H. and Funabiki, N. (2015) 'Performance of implementation IBR-DTN and Batman-Adv routing protocol in wireless mesh networks', *EMITTER International Journal of Engineering Technology*, Vol. 3, No. 1, pp. 19-37

Zeppelin (2017) *Apache Zeppelin* [online] http://zeppelin.apache.org. (accessed 12 April 2017).

Figure 1 IoT Reference Model

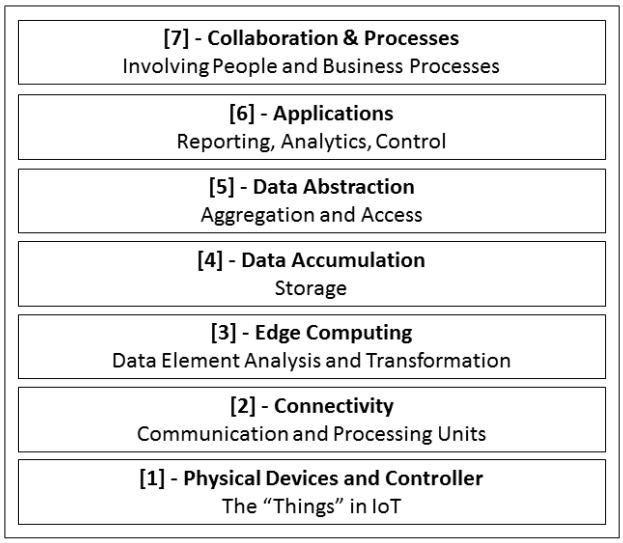


Figure 2 System Design.

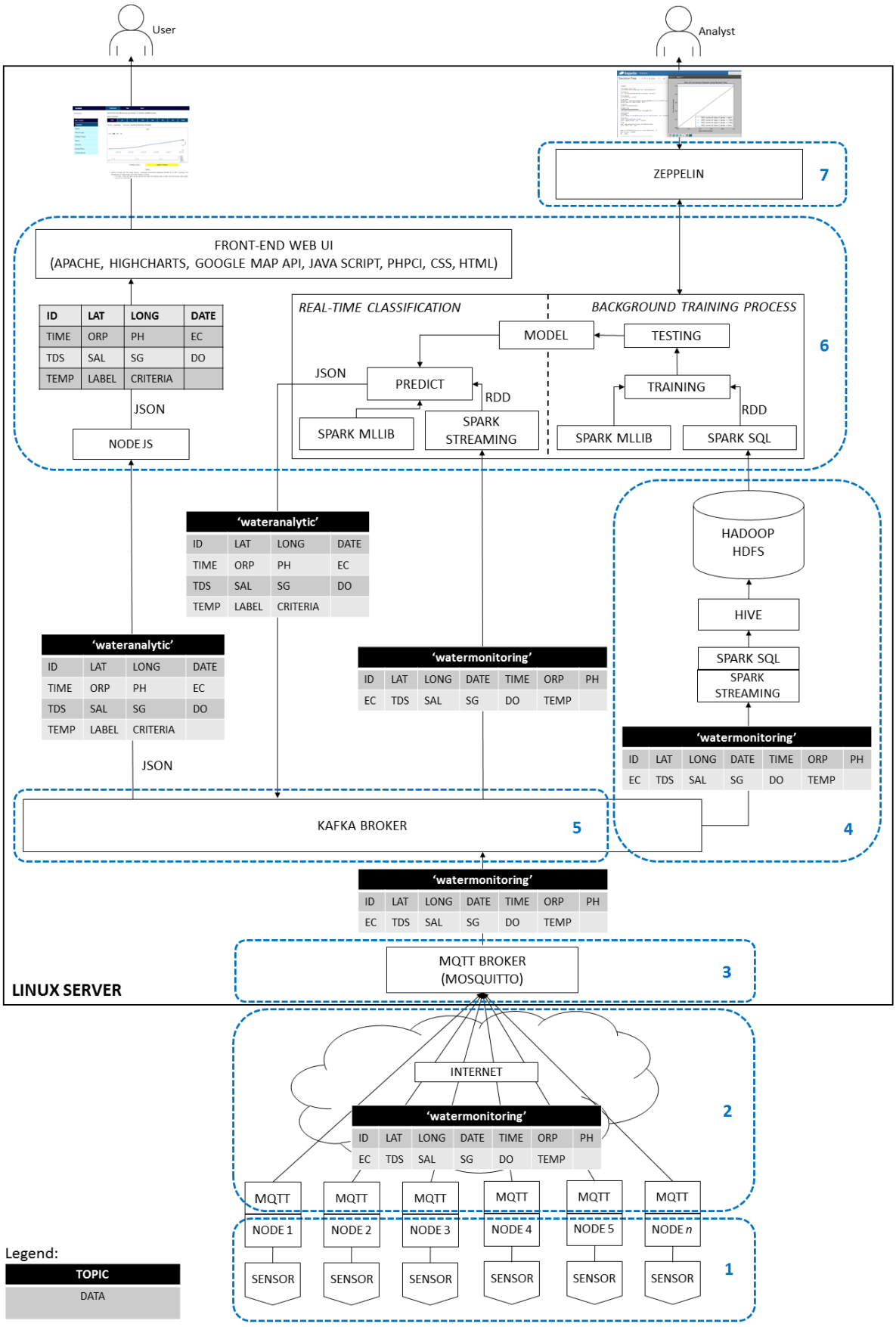


Figure 3 Data Flow in Visualisation.

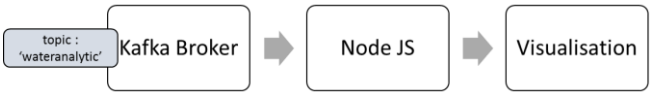


Figure 4 Visualisation of Node Location.

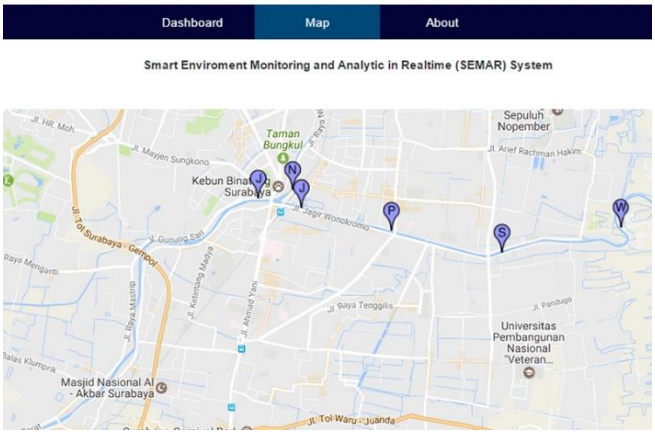


Figure 5 Visualisation of Water Data Sensor and Information of Pollution Index.

Figure 6 ROC of laboratory test dataset using Linear Support Vector Machine and Decision Tree.

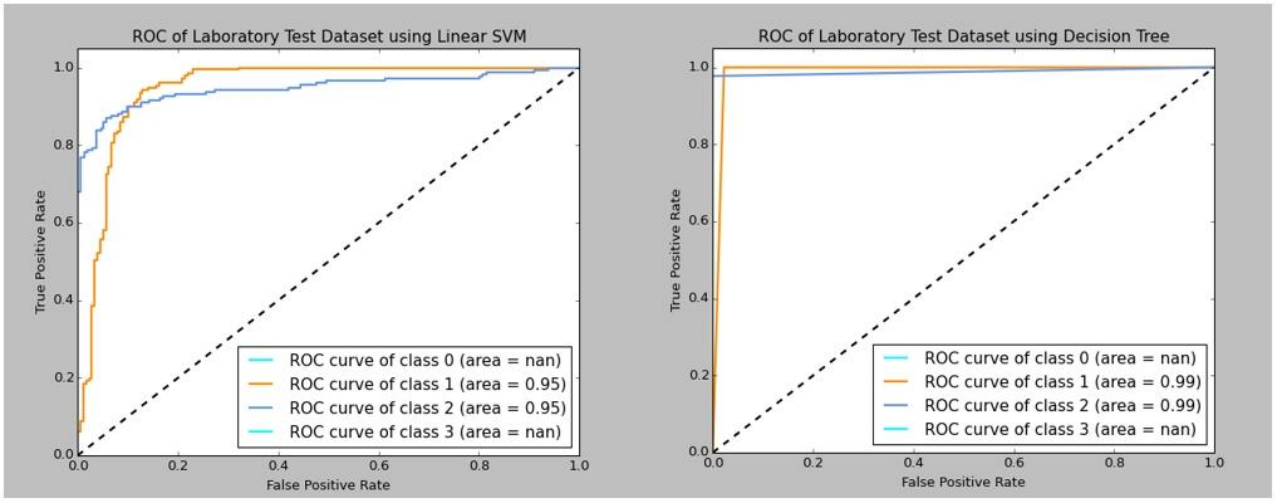


Figure 7 ROC of live sensor dataset using Linear Support Vector Machine and Decision Tree.

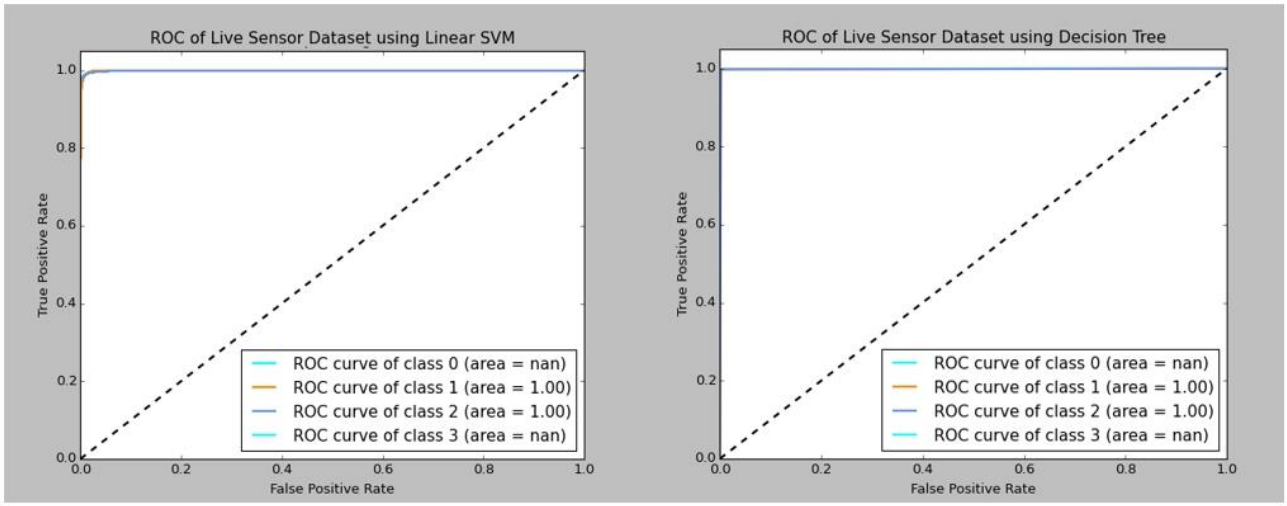


Figure 8 Server Processing Time

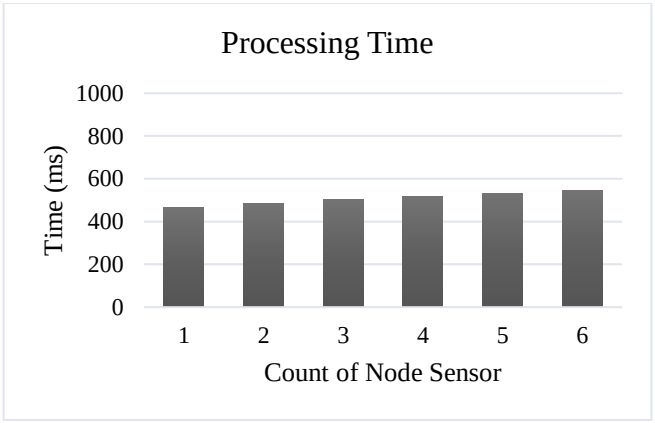


Table 1 Pollution Index Category.

No.	Pollution Index	Categories
1	$0 \leq PI_j \leq 1.0$	Fulfil Standard
2	$1.0 < PI_j \leq 5.0$	Lightly Polluted
3	$5.0 < PI_j \leq 10.0$	Polluted
4	$PI_j > 10.0$	Heavy Polluted

Table 2 Confusion Matrix of laboratory test dataset uses Linear Support Vector Machine.

		Predicted Class			
		0	1	2	3
Actual Class	0	0	0	0	0
	1	0	206	6	0
	2	0	19	159	0
	3	0	0	0	0

Table 3 Confusion Matrix on laboratory test dataset uses Decision Tree.

		Predicted Class			
		0	1	2	3
Actual Class	0	0	0	0	0
	1	0	212	0	0
	2	0	2	176	0
	3	0	0	0	0

Table 4 Confusion Matrix of live sensor dataset uses Linear Support Vector Machine.

		Predicted Class			
		0	1	2	3
Actual Class	0	0	0	0	0
	1	0	19770	107	0
	2	0	182	25338	0
	3	0	0	0	0

Table 5 Confusion Matrix of live sensor dataset uses Decision Tree.

		Predicted Class			
		0	1	2	3
Actual Class	0	0	0	0	0
	1	0	19859	18	0
	2	0	16	25504	0
	3	0	0	0	0

Table 6 Comparison of two algorithms on laboratory test and live sensor dataset.

Features	Dataset	Algorithm	Mislabel	Accuracy	MSE
pH, TSS, DO, Temp, Turbidity	Laboratory test	Linear Support Vector Machine	25 / 390	0.935897	0.0641
		Decision Tree	2 / 390	0.994872	0.0051
	Live Sensor	Linear Support Vector Machine	289 / 45397	0.993634	0.0064
		Decision Tree	34 / 45397	0.999251	0.0007