EPI International Journal of Engineering
Volume 3, Number 2, August 2020, pp. 119-125
DOI: 10.25042/epi-ije.082020.05

*p*ISSN 2615-5109
*e*ISSN 2621-0541

Monitoring and Predicting Water Quality in Swimming Pools

Apriandy Angdresey[a], Lanny Sitanayah[b,*], Vandri Josua Abram Sampul[c]

[a]Department of Informatics Engineering, Faculty of Engineering, Universitas Katolik De La Salle, Manado - Indonesia.
Email: aangdresey@unikadelasalle.ac.id
[b]Department of Informatics Engineering, Faculty of Engineering, Universitas Katolik De La Salle, Manado - Indonesia.
Email: lsitanayah@unikadelasalle.ac.id
[c]Department of Informatics Engineering, Faculty of Engineering, Universitas Katolik De La Salle, Manado - Indonesia.
Email: 16013015@unikadelasalle.ac.id

Abstract

Water quality in public swimming pools affects human health. While changing the water too soon is wasteful, postponing changing the dirty water is not hygiene. In this paper, we propose an Internet of Things-based wireless system to monitor and predict water quality in public swimming pools. Our system utilizes an Arduino Uno, an ESP8266 ESP-01 WiFi module, a DS18B20 temperature sensor, a pH sensor, and a turbidity sensor. We predict the water quality using a data mining prediction model, namely the decision tree Iterative Dichotomiser 3 algorithm. We show by experiment that our sensor node and the wireless monitoring system work correctly. We also show by simulation using Weka that we can get 100% accuracy with a kappa statistical value of 1 and 0% error rate.

Keywords: Data mining; decision tree; iterative dichotomiser 3; Internet of Things

1. Introduction

Swimming is a kind of sport fancied by many people; not only children, but also adults. This sport makes our body healthy, because when we swim, we use almost all muscles in our body. Swimming is not only a sport, but also a recreational activity. Nowadays, there are many public swimming pools available for recreation or competition. As the pools are used by many people, water cleanliness triggers health issues. Swimming in a dirty pool can cause infectious diseases, such as skin diseases, diarrhea, eye irritation, and respiratory tract irritation.

In swimming pools, we can control disease-causing bacteria by adding chlorine in the water. Water acidic or the pH level is very important to determine how effective the disinfectant can kill germs. However, people can get eye or respiratory tract irritation due to the vapor from chloramine that is formed by chlorine and sweat or urine. Therefore, changing pool water regularly is necessary to improve water quality and hygiene. A simple solution is using a fixed schedule to change the water, for example once a week or every two weeks. However, if the water is still clean, it will be wasteful. If the water is too dirty, it is not hygiene until the next change schedule. Another solution is measuring the pH level and turbidity of the water manually, but this too is neither cost effective nor efficient as it may require several personnel visits to the pool.

Monitoring and predicting water quality in swimming pools can be done remotely thanks to the Internet of Things. The Internet of Things offer new platforms that enable physical objects equipped with sensor devices – with or without batteries [1] – to communicate and exchange information wirelessly without human intervention [2]. In this paper, we develop an Internet of Things-based wireless system to monitor and predict water quality in public swimming pools. Our contributions in this paper are two-folds:

1) Firstly, we design and implement a wireless system to monitor water quality in public swimming pools using an Arduino Uno, an ESP8266 ESP-01 WiFi module, a DS18B20 temperature sensor, a pH sensor, and a turbidity sensor.
2) Then, we utilize a data-mining prediction model to predict the quality of water in public swimming pools using the decision tree Iterative Dichotomiser 3 (ID3) algorithm.

The rest of this paper is organized as follows. Section 2 discusses the related works. Section 3 shows our system framework, such as hardware and software design and implementation. Section 4 describes our methods for water quality prediction. Section 5 reports our experiments and simulation results. Finally, Section 6 concludes this paper and suggests some future directions.

*Corresponding author. Tel.: +62-812-4365-9655
Universitas Katolik De La Salle
Manado, Indonesia

2. Related Works

The Internet of Things (IoT) enable physical objects equipped with smart sensors, *i.e.* from gas sensors [3] to medical sensors, to transmit data using wireless technology. One of the implementations of the Internet of Things is a Wireless Sensor Network-based monitoring and control system based on the use of electrical energy [4]. In this study, wireless sensor nodes are designed to measure electrical parameters of Alternating Current (AC) as effective voltage, effective current, active power, apparent power, power factor and total electrical energy consumption using the ESP8266 module as a link with WiFi. Power consumption monitoring through the Internet can show several electrical parameters with the same data from the data logger recap taken from an SD-Card installed in a wireless sensor node through an Android application and web browser.

In [5], the authors monitor the quality of lake water based on the parameters of acidity (pH), turbidity, dissolved oxygen (DO) and water temperature to provide an early warning when the lake water quality is approaching a given threshold, so that the loss of fish farmers in the lake can be minimized. The data of water quality measured by existing sensors is collected in a data logger and forwarded using a cellular network. The data is stored and displayed on a website and can be accessed from anywhere.

By using available and relevant data, data mining can generate several models to identify patterns among attributes that exist in a data-set. Some of the patterns can be predictive such as predicting the value or results that will occur on certain attributes. As in [6], the authors identify and predict the sex of tarantula using training data taken for almost one year. In addition, data mining can be implemented to conduct opinion mining, such as text document that crawls in social media that contains comments, criticism and reviews about presidential candidates of Republic Indonesia in 2019 [7].

There are several algorithms that can be applied in processing data mining; one of which is decision tree. Most of data mining algorithms are supervised learning. Decision tree uses a hierarchical structure for supervised learning. This algorithm performs the learning process based on the value of the target variable associated with the value of the predictor variables. By using this algorithm, the area of decision making, which was previously complex and very global, can be changed to be simple and specific.

There are several implementations of decision tree algorithms, such as in [8], the authors investigate an Android malware and detect it using decision tree algorithms. In [9], the study is about the influence on share price in accordance with financial fundamentals of a company. This paper builds a stock investment using the decision tree model through the ID3 algorithm. The results can provide decision support for future investments through the analysis of the current financial and market situation. The ID3 algorithm is also used in expert systems, such as in [10], where they develop an expert system that can predict whether a person is addicted to being vulnerable to drugs or not. Hence, they can control and be aware of any drug abuser as they can conduct the test repeatedly to cure them without hesitation.

3. System Design and Implementation

The wireless system for water quality monitoring in swimming pools is depicted in Fig. 1. Our system consists of one sensor node that periodically gathers data from a swimming pool and sends the data wirelessly via the Internet to the web server to be processed further. A user can then access the information using a web browser.

The main components of our sensor node are an Arduino Uno [11], an ESP8266 ESP-01 WiFi module [12], and three sensors, *i.e.* a DS18B20 temperature sensor, a pH sensor and a turbidity sensor. The three sensors and ESP8266 ESP-01 are connected to Arduino via wired connections. The sensor node design is shown in Fig. 2. Arduino periodically collects data from the three sensors and utilizes the WiFi module to send data wirelessly to the web server. We show the sensor node design in Fig. 2, the real image of the sensor node in Fig. 3 and the list of our hardware and software used in Table 1.

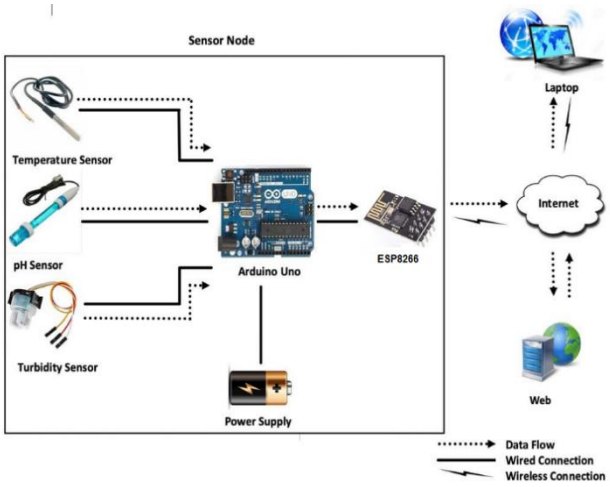


Figure 1. The monitoring system

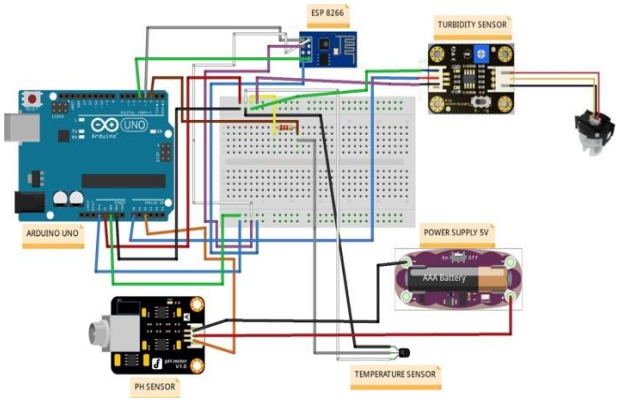


Figure 2. The sensor node design

Table 1. Hardware and software

Hardware/Software	Purpose
Arduino Uno	Collect, transform and send data.
ESP8266 ESP-01	WiFi module.
pH Sensor	Acidity sensor.
Turbidity Sensor	Turbidity sensor.
DS18B20 Sensor	Temperature sensor.
Laptop/Mobile Device	Access data from the web server.
Arduino Program	Collect, transform and send data.
PHP Program	Process data.

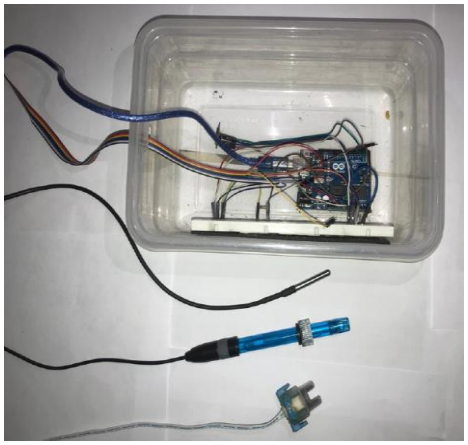


Figure 3. A sensor node consists of an Arduino Uno, a pH sensor, a turbidity sensor, a temperature sensor and an ESP8266 ESP-01

We implemented two programs to be used in our system for data collection, data transmission, and data processing. The first program is implemented using Arduino IDE and uploaded to the Arduino board. This program reads data from the three sensors, transforms and sends them using the WiFi module. The second program is implemented using Hypertext Preprocessor (PHP) for the web server. This program processes the data using machine learning techniques to produce useful information for users who can access it from a web browser.

Figure 4 shows processes in the client-side (sensor node). After system initialization, if an Internet connection is available, the sensor node gets readings from the three sensors, i.e. the temperature, pH, and turbidity sensors. Then, the raw data is transformed using predefined categories. For example, there are three categories for acidity, namely acidic, neutral, and alkaline. The two categories for turbidity are turbid and non-turbid, while temperature is categorized into hot, normal, and cold. The transformed data is sent to the server wirelessly, then the buffer is cleared. These processes occur repeatedly in a loop.

Figure 5 shows processes in the server-side, which also occur in a loop. Firstly, if the server is connected to the Internet, it can receive data sent by the sensor node. The data is then processed using the decision tree Iterative Dichotomiser 3 algorithm and stored in the database.

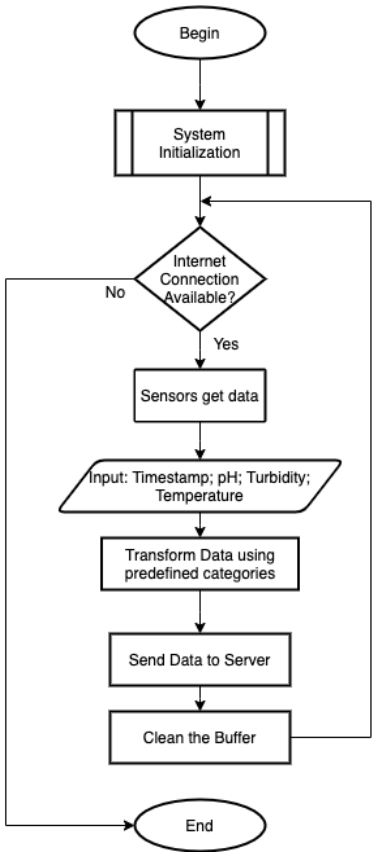


Figure 4. Client flowchart

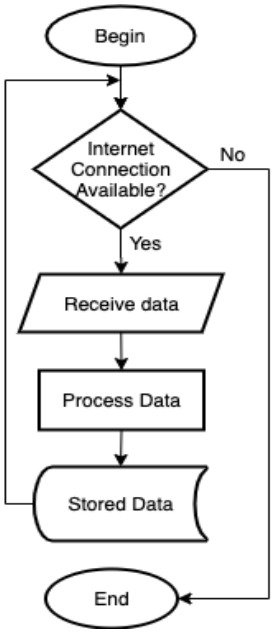


Figure 5. Server flowchart

4. Prediction Methods

In this section, we will discuss about the prediction methods, *i.e.* the decision tree methods with the ID3 algorithm on training data to predict water quality. The ID3 algorithm is an algorithm used to construct a decision tree that uses a hierarchical structure for supervised learning. The process of constructing the decision tree is done recursively.

The classification process is divided into two phases, namely learning and test. In the learning phase, most of the data that has been known its class of data are fed to model estimates. Moreover, in the test phase, the model formed from the learning phase is tested with most of other data to determine the accuracy of the model. This model is used for our prediction.

Historical data is presented in a table with attributes and records. Attributes declared as a parameter is created from the training data as a criterion in forming the tree, and the attribute values are called instances. In this study, we consider acidity, turbidity and temperature of water. Historical data is taken by existing sensors and is collected directly to web server. After that, the historical data is cleared and transformed. The value of each data obtained is numeric, so we change the data by categorizing it to show several different conditions. In this case, for acidity we have three categories, namely acidic (pH 0-6), neutral (pH 6.1-7.9) and alkaline (pH 8-14). Turbidity is divided into clear and cloudy. And for temperature, there are three categories, namely hot (less than 25 degrees), normal (25-35 degrees) and cold (more than 35 degrees). Entropy, as shown in Equation 3, defines the value of information gain, which is used to measure the effectiveness of an attribute in data classification. Entropy, defined in Equation 2, is the sum of probability of each label times the log probability of that same label. P_i, as shown in Equation 1, is the ratio of class i in the set S which is a set of attributes that are classified.

$$P_i = \frac{\sum_k^m X_k \in (i)}{S} \quad (1)$$

$$Entropy(S) = \sum_i^n -P_i Log_2(P_i) \quad (2)$$

$$Gain(S, A) = Entropy(S) - \sum_{\upsilon \in Values(A)} \frac{S\upsilon}{St} EntropySv \quad (3)$$

Where A is an attribute, S_v is declared number of samples for the value of v, and St is the total number of data samples. Meanwhile, entropy S_v is the entropy for samples that has value v. The following is an example to calculate entropy and gain. Suppose that the existing historical data have attributes such as Weather, Temperature, Humidity and Windy. Each attribute has a value. There are two classes: "play" and "not play". If the number of "play" class is 10 and the number of "not play" class is 4, then

$$Entropy(s) = (-(10/14)\ log_2(10/14) + (-(4/10)\ log_2(4/10)) = 0.863120569 \quad (4)$$

After getting the entropy of the whole case, one needs to perform analysis on each of the attributes and values and calculate the entropy. After getting the entropy value, the next step is to calculate the information gain value of each variable. *Gain(Weather)* = 0.863120569 - ((4/10) 0 + (5/14) 0.721928095 + (5/14) 0.970950594) = 0.258521037. Similarly, one needs to calculate *Gain(Temperature)*, *Gain(Humidity)*, and *Gain(Windy)*. The largest gain value will be the root node.

Furthermore, the quality of the classification rule will be evaluated using accuracy. Accuracy is the percentage of tuples that are covered and properly classified.

5. Evaluation and Result

We evaluate the wireless monitoring system by firstly showing that the sensor node and the system work correctly. Then, we show the result of our ID3 algorithm's simulation using data gathered from the sensor node.

(a)

(b)

Figure 6. Clean water test

(a)

(b)

Figure 7. Hot coffee test

5.1. Hardware evaluation

To test whether the sensor can sense correctly, firstly we test it by using spring water. We put spring water, which has neutral acidity, clear turbidity, and normal temperature, in a container as shown in Fig. 6(a). Then, we let the sensor detect the water quality and send the data to the web server to be processed and displayed at the web browser as depicted in Fig. 6(b). The second test uses hot coffee, which is acidic, not clear (cloudy), and hot, as shown in Fig. 7(a). This test too can be displayed correctly at the web browser as depicted in Fig. 7(b).

5.2. System evaluation

We evaluate our wireless monitoring system to show that the sensor node is able to capture acidity, turbidity and temperature data from a swimming pool, transform, send, and store the data correctly in our database. In our evaluation, the sensor node senses the environment every 10 minutes. However, for a better presentation in this paper, we only capture several chunks of data from the database that show different monitoring conditions. Note that in this system, the sensor node does not send numerical data, but only the transformed data.

Figure 8 shows the acidity transformed data, Figure 9 shows the turbidity transformed data, and Figure 10 shows the temperature transformed data, where one time unit is equivalent to 10 minutes. As we can retrieve this data from the database, it means our wireless monitoring system is able to collect data from the three sensors, transform, and send the data to the web server correctly. Table 2 shows the transformed data and the classes (clean and dirty) used in prediction.

Table 2. Transformed data

No.	Acidity	Turbidity	Temperature	Class
1	Neutral	Clear	Normal	Clean
2	Neutral	Clear	Normal	Clean
3	Neutral	Clear	Normal	Clean
…	…	…	…	…
21	Acidic	Clear	Normal	Clean
22	Acidic	Clear	Normal	Clean
79	Alkaline	Clear	Normal	Clean
80	Neutral	Clear	Normal	Clean
…	…	…	…	…
105	Alkaline	Clear	Normal	Clean
106	Acidic	Cloudy	Hot	Dirty
107	Acidic	Cloudy	Hot	Dirty
…	…	…	…	…
147	Acidic	Clear	Cold	Dirty
148	Acidic	Cloudy	Hot	Dirty
…	…	…	…	…
222	Alkaline	Clear	Cold	Dirty
223	Alkaline	Clear	Cold	Dirty
224	Alkaline	Cloudy	Cold	Dirty
…	…	…	…	…
340	Alkaline	Cloudy	Normal	Dirty
341	Alkaline	Cloudy	Normal	Dirty
342	Alkaline	Cloudy	Normal	Dirty

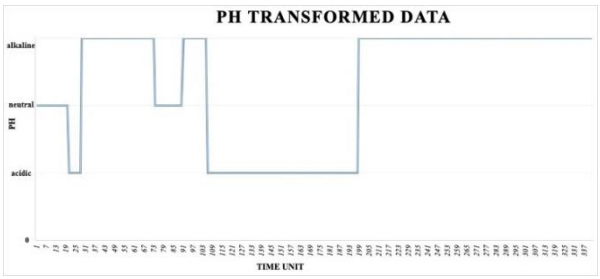


Figure 8. Acidity transformed data

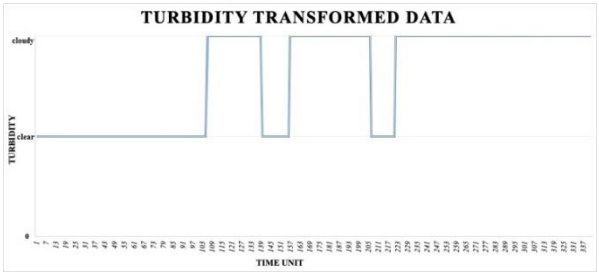


Figure 9. Turbidity transformed data

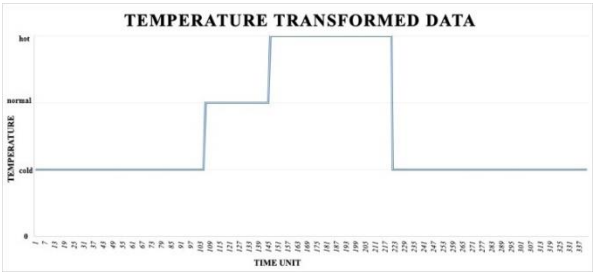


Figure 10. Temperature transformed data

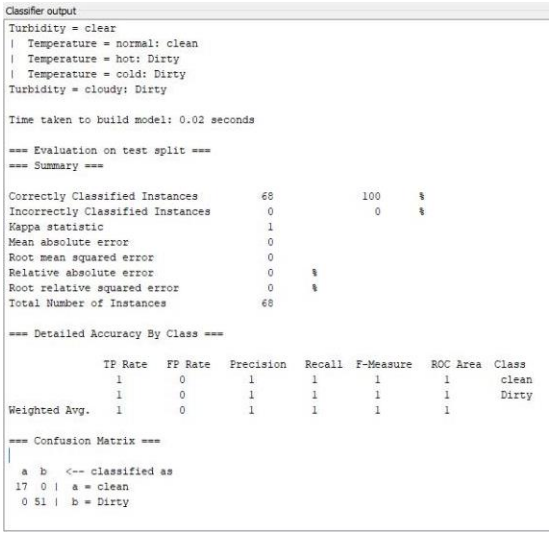
5.3. ID3 simulation

In this phase, we conduct two trials. In the first trial, we build the water quality monitoring application. Users can see the sensor data in real time and make predictions using ID3, which is shown in Fig. 11.

Figure 11. The water quality monitoring application

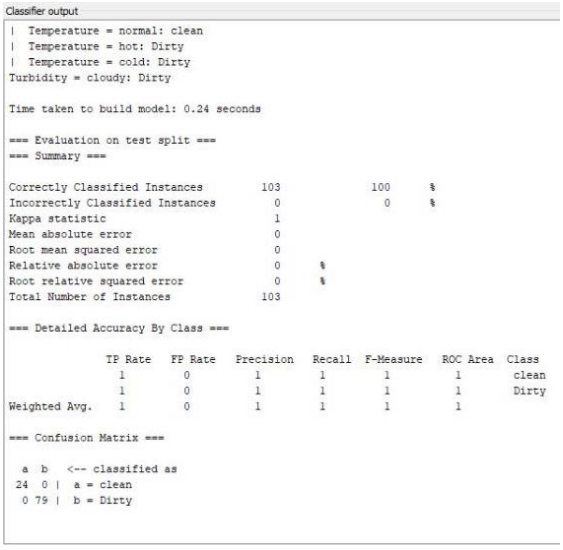
The second trial use Weka to conduct simulations on learning data that we get to calculate the accuracy and kappa values. We divide the simulations into three different experiments. In the first experiment, we split the data into 80% learning data and 20% test data. In the second experiment, we split the data into 70% and 30%, then in the third experiment, we split it into 60% and 40%.

Figure 12 shows the results of the first experiment, Figure 13 shows the second experiment, and Figure 14 shows the third one. From these results, it can be seen that the accuracy value obtained from these three experiments is 100% with a kappa statistical value of 1 and 0% error rate. Figure 15 shows the comparison of execution time from the first experiment to the third experiment, where the second and the third experiments have better execution time.



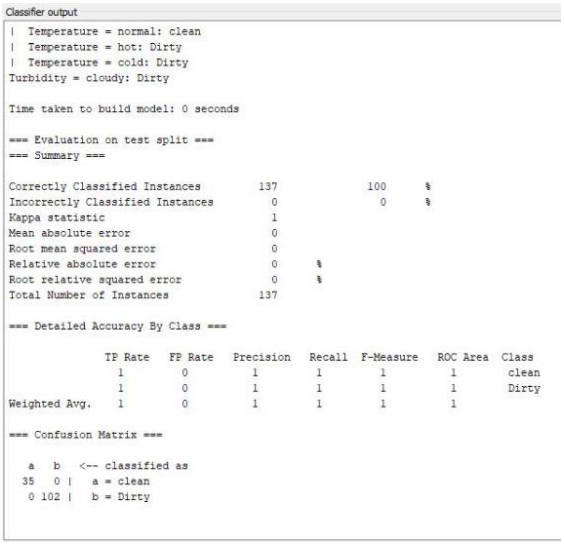
```
Classifier output
Turbidity = clear
|  Temperature = normal: clean
|  Temperature = hot: Dirty
|  Temperature = cold: Dirty
Turbidity = cloudy: Dirty

Time taken to build model: 0.02 seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances          68               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0
Root mean squared error                  0
Relative absolute error                  0      %
Root relative squared error              0      %
Total Number of Instances               68

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        clean
                 1         0          1         1         1          1        Dirty
Weighted Avg.    1         0          1         1         1          1

=== Confusion Matrix ===

  a  b   <-- classified as
 17  0 |  a = clean
  0 51 |  b = Dirty
```

Figure 12. Experiment 1



```
Classifier output
|  Temperature = normal: clean
|  Temperature = hot: Dirty
|  Temperature = cold: Dirty
Turbidity = cloudy: Dirty

Time taken to build model: 0.24 seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances         103               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0
Root mean squared error                  0
Relative absolute error                  0      %
Root relative squared error              0      %
Total Number of Instances              103

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        clean
                 1         0          1         1         1          1        Dirty
Weighted Avg.    1         0          1         1         1          1

=== Confusion Matrix ===

  a  b   <-- classified as
 24  0 |  a = clean
  0 79 |  b = Dirty
```

Figure 13. Experiment 2



```
Classifier output
|  Temperature = normal: clean
|  Temperature = hot: Dirty
|  Temperature = cold: Dirty
Turbidity = cloudy: Dirty

Time taken to build model: 0 seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances         137               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0
Root mean squared error                  0
Relative absolute error                  0      %
Root relative squared error              0      %
Total Number of Instances              137

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        clean
                 1         0          1         1         1          1        Dirty
Weighted Avg.    1         0          1         1         1          1

=== Confusion Matrix ===

   a   b   <-- classified as
  35   0 |   a = clean
   0 102 |   b = Dirty
```

Figure 14. Experiment 3

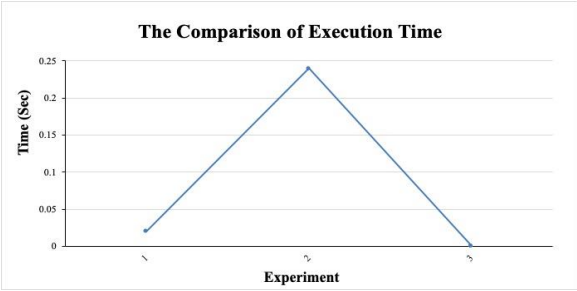


Figure 15. Execution time comparison

6. Conclusion and Future Work

In this paper, we propose a wireless solution based on the Internet of Things to monitor and predict water quality in public swimming pools. Our system consists of one sensor node that periodically sends temperature, pH, and turbidity conditions to a web server for further processing. We utilize the Decision Tree method, namely Iterative Dichotomiser 3, to predict the water quality. We show by experiment that our sensor node and the wireless monitoring system work correctly. Then, based on the Weka simulations, we show that the experiments achieve 100% accuracy with a kappa statistical value of 1 and 0% error rate.

Currently, our monitoring system has only one sensor node. In the future, we plan to add more sensor nodes and deploy them at different places in a pool. We also plan to utilize more sensors, so we can get more variables for prediction. As our current system can only monitor and predict water quality, we would like to incorporate automation capability into the system. For example, adding chlorine automatically to improve the pool's water quality.

References

[1] J. Hester, N. Tobias, A. Rahmati, L. Sitanayah, D. Holcomb, K. Fu, W. Burleson, and J. Sorber. Persistent clocks for batteryless sensing devices. *ACM Transactions on Embedded Computing Systems*, 15:1–28, Aug. 2016.

[2] F. Balo. Internet of things: A survey. International Journal of Applied Mathematics, Electronics and Computers, pages 104–110, Dec. 2016.

[3] J.B. Sanger, L. Sitanayah, and V.D. Kumenap. Detection system for cigarette smoke. In *4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, pages 145–149, 2019.

[4] I. Giriantari, I. Eka Putra and L. Jasa. Monitoring penggunaan daya listrik sebagai implementasi internet of things berbasis wireless sensor network. *Majalah Ilmiah Teknologi Elekt*ro, 16(3):50–55, 2017.

[5] D. Sudirman and N. Hidayatullah. Aplikasi sistem monitoring kualitas air danau sebaga mitigasi kebencanaan berbasis IoT. *Jurnal Poli-Teknologi*, 18(2), 2019.

[6] A. Angdresey and M. Wongkar. Identification of the reproductive apparatus of tarantula genus brachypelma using linear discriminant analysis method. *In International Conference on Electrical Engineering and Computer Science (ICECOS)*, pages 409–414, 2018.

[7] M. Wongkar and A. Angdresey. Sentiment analysis using naïve bayes algorithm of the data crawler: Twitter. *In Fourth International Conference on Informatics and Computing (ICIC)*, pages 1–5, 2019.

[8] A. Utku, I.A. Do ̆gru, and M.A. Akcayol. Decision tree based android malware detection system. *In 26th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4, 2018.

[9] C. Chen. The apply of ID3 in stock analysis. *In 6th International Conference on Computer Science Education (ICCSE)*, pages 24–27, Aug. 2011.

[10] Kaur and R.K. Bawa. Implementation of an expert system for the identification of drug addiction using decision tree ID3 algorithm. *In 3rd International Conference on Advances in Computing, Communication Automation (ICACCA) (Fall)*, pages 1–6, Sep. 2017.

[11] Arduino. Available at https://components101.com/ microcontrollers/arduino-uno. [22 February 2020].

[12] Getting Started With the ESP8266 ESP-01. Available at https://www.instructables.com/id/Getting-Started-With-the-ESP8266-ESP-01/. [22 February 2020].