EPI International Journal of Engineering
Volume 9, Number 1, February 2026, pp. 1-9
DOI: 10.25042/epi-ije.022026.01

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

# An Internet of Things-Based Trash Can Monitoring System

Lanny Sitanayah[a,*], Felicia Kusuma[b], Steven Pandelaki[c]

[a]Department of Informatics Engineering, Faculty of Engineering, Universitas Katolik De La Salle, Manado - Indonesia.
Email: lsitanayah@unikadelasalle.ac.id.
[b]Department of Informatics Engineering, Faculty of Engineering, Universitas Katolik De La Salle, Manado - Indonesia.
Email: 18013106@unikadelasalle.ac.id
[c]Department of Informatics Engineering, Faculty of Engineering, Universitas Katolik De La Salle, Manado - Indonesia.
Email: spandelaki@unikadelasalle.ac.id

## Abstract

Effective waste management remains a persistent challenge for modern urban centers, as the optimization of collection and disposal operations is fundamentally linked to environmental integrity and public health outcomes. In addition to the physical accumulation of waste, a significant systemic challenge is the absence of real-time data regarding the status of trash can, which tend to change from time to time. This research addresses the critical challenge of urban waste management by developing a resource-efficient, IoT-enabled monitoring system optimized for low-power hardware. While high-performance machine learning models achieve high accuracy, their computational demands often exceed the capabilities of distributed IoT nodes. This study fills that research gap by implementing a Gaussian Naïve Bayes algorithm on a NodeMCU platform, prioritizing computational efficiency and scalability in memory-constrained environments. Integrating ultrasonic and gas sensors to monitor waste volume and odor, the system utilizes a systematic data collection and labeling protocol to monitor trash can conditions. Experimental results demonstrate that the Naive Bayes approach provides a lightweight yet robust solution, maintaining reliable performance (~80% accuracy) in distinguishing trash can conditions while maintaining a light computational load (processing time and memory consumption).

*Keywords:* Classification; data mining; gas sensor; naive bayes; ultrasonic sensor

## 1. Introduction

Urban waste management has long been a challenging task for modern smart cities, as the operational efficiency of waste management can directly impact public health and environmental quality [1]. Global municipal solid waste production reached 2.01 billion tonnes in 2016 and is projected to escalate to 3.4 billion tonnes by 2050 [2]. Beyond the obvious problem of waste accumulation, a real issue lies in the lack of real-time information on the status of waste disposal points, which can change dynamically. Currently, in waste management, city authorities operate with a lack of accurate real-time data on the quantity and rate of biochemical decomposition of organic matter at each waste collection point. Without such detailed real-time data, accurately prioritizing cleanup becomes impossible.

Conventional waste collection methods further exacerbate these inefficiencies. They rely heavily on manual, periodic inspections by sanitation workers moving from one bin to another, which is both inefficient and costly. This "blind" collection strategy wastes fuel and labor by visiting nearly empty bins, while at the same time there are overflowing bins in crowded areas, emitting foul odors due to the rapid decomposition of organic matter [3]. This lack of situational awareness is not just a logistical hassle; it represents a fundamental flaw in urban infrastructure – the absence of an objective, parameter-based early warning system that can distinguish between bins that are not yet in need of emptying and those that are "Really Need to Be Cleaned."

To bridge this critical information gap, the implementation of Internet of Things (IoT) technology is no longer an optional enhancement but a technical urgency. IoT is a concept that has the goal to further expand Internet connectivity, that is connected people, systems, and information resources continuously, which allows communication and collaboration with hardware and can display real world things on the Internet [4]. With IoT, all equipments or objects around us can be connected to each other via the Internet, making it easier for users to use and communicate wherever the user is. By integrating ultrasonic sensors to monitor physical volume and gas sensors to detect decomposition levels, a robust data-driven framework can be established [5]. Such a system enables autonomous classification, effectively shifting the urban monitoring paradigm from a reactive, schedule-

---

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

based approach – which often fails to adapt to sudden surges in waste – to a proactive, data-centric model. This digital transformation allows for the synchronization of physical cleanup activities with actual waste generation patterns, ensuring that interventions occur precisely when and where they are most needed.

Building upon this technological necessity, this research aims to develop an intelligent trash bin monitoring system designed to classify cleaning priorities with high precision using the Gaussian Naïve Bayes algorithm. While many high-level machine learning models exist, the specific contribution of this study lies in evaluating the efficiency of Gaussian Naïve Bayes in an IoT environment with resource-constrained hardware. Most IoT nodes are resource-constrained devices with limited processing power and memory; therefore, finding an algorithm that balances classification accuracy with extreme computational efficiency is vital. By validating this approach, this research provides a scalable and practical solution for sanitation managers to optimize cleaning routes and schedules in real-time, ultimately reducing operational costs and enhancing the livability of the urban environment.

## 2. Literature review

The rapid evolution of modern technology has established the Internet of Things (IoT) as a transformative breakthrough that is now inextricably linked to daily human activities [6]. At its core, the IoT framework involves the integration of electronic components into everyday objects, effectively converting them into "smart devices" capable of connecting, communicating, and exchanging complex data packets over a standard Internet connection. A defining characteristic of these devices is their capacity for Machine-to-Machine (M2M) communication, which allows hardware to interact with minimal human intervention, thereby streamlining operational efficiency [7].

The architecture of an IoT-enabled object typically centers around a microcontroller, which serves as the local intelligence hub. This controller manages one or more sensors that capture environmental variables – such as fill levels in a bin or ambient gas concentrations – and transmits this raw information to a centralized application where it can be monitored by users in real-time. This interaction is inherently bidirectional; beyond passive observation, users can exert direct influence over the object by issuing remote commands through a digital interface. These instructions are then executed by actuators, which serve as the physical component responsible for translating digital signals into mechanical action.

IoT-integrated intelligent bins have been shown to significantly outperform traditional waste management by utilizing real-time monitoring to optimize logistics. It was reported in [8] that the 6-month implementation in urban setting led to a 50% increase in collection frequency, which directly contributed to a 66.7% reduction in missed collections and a substantial 80% decrease in bin overflow incidents. These operational improvements resulted in a 20% improvement in fuel consumption and a 15% reduction in total operating costs.

Previous studies in waste monitoring automation, such as in [9] use ultrasonic sensors to detect waste levels in real-time. Specifically, the ultrasonic sensor is mounted within the bin to measure the distance between the sensor and the accumulated waste. In addition to employing ultrasonic sensors for fill-level monitoring, the studies in [10], [11], and [12] incorporate gas sensors for odor detection. Other kinds of sensors used are rain sensors [11] and temperature and humidity sensors [12] to monitor environmental conditions. In [13], load sensors are used to measure weight, while in [14], other than load sensors, infra-red sensors are added to detect any movement.

In terms of algorithms used, the IoT-based system in [15] utilizes smart bins and sensors to collect granular data on waste volume, disposal trends, and classification. The system then applies a Naive Bayes classifier to analyze historical data and forecast future waste patterns with high precision. In [16], the authors developed an IoT-based smart waste management framework that utilizes a multi-sensor network to monitor real-time environmental data, including bin capacity, humidity, temperature, and ambient light, with data accessible via a mobile application. To classify bin status, the study evaluated three machine learning models – Support Vector Machine (SVM), K-Nearest Neighbors (K-NN), and Decision Tree (DT) – using a dataset of 150 samples. Their experimental results identified SVM as the most effective algorithm for this application, achieving a high classification accuracy of 92.3%. In [17], the authors compared Random Forest, K-NN, Artificial Neural Network (ANN), SVM, and Naïve Bayes. The results show that Random Forest, KNN, and SVM have better accuracy (over 90%) than Naive Bayes and ANN.

Naïve Bayes is a supervised learning algorithm, where this algorithm is used for classification in machine learning and is based on Bayes' theorem [18]. Naïve Bayes works on probability distributions. Naïve Bayes can help with fast classification and prediction with better accuracy. The advantage of Naïve Bayes is that it only requires a small amount of training data to be used for classification. While existing literature extensively explores high-accuracy machine learning models such as Random Forest and Support Vector Machines for waste management, these approaches often overlook the hardware constraints inherent in distributed IoT networks with low-resource nodes. Our study fills this void by investigating the computational efficiency of the Gaussian Naïve Bayes algorithm specifically for power- and memory-limited environments like the NodeMCU. By prioritizing a lightweight probabilistic model over resource-heavy alternatives, this research demonstrates that significant operational intelligence can be achieved without the high energy consumption or hardware costs typically associated with advanced neural networks, thereby offering a more scalable solution for large-scale urban deployments.

## 3. Method

In Naïve Bayes, the formula for determining the probability of event A after event B has occurred, is:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad (1)$$

and

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} \quad (2)$$

From conditional probabilities:

$$P(A \cap B) = P(A)P(B|A) \quad (3)$$

and

$$P(A \cap B) = P(B)P(A|B) \quad (4)$$

From the two equations above, the Naïve Bayes equation is:

$$P(A|B) = \frac{P(B|A)}{P(B)} \quad (5)$$

where
$A$ = results or hypotheses
$B$ = features in the data

Gaussian Naïve Bayes is the last step of Naïve Bayes to find out the results of a data test model by taking the value of the opportunity from the training data. Formulas from Gaussian Naïve Bayes is:

Determine the mean ($\bar{x}$) and variance ($\sigma$):

$$Mean\ (attribute) = \frac{\Sigma x_i}{n} \quad (6)$$

where
$\Sigma x_i$ = the sum of the data from each attribute
$n$ = total amount of data

$$Variance(attribute) = \frac{\Sigma (x_i - \bar{x})^2}{n-1} \quad (7)$$

where
$\bar{x}$ = the mean value of each attribute

Finding the probability value of each attribute based on class using the Gaussian distribution formula:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\bar{x})^2}{2\sigma^2}} \quad (8)$$

where
$f(x)$ = new values based on each class

Finding the posterior probability value:

$$posterior(class) = p(class) * p(attribute\ 1|class) * \ldots p(attribute\ n|class) \quad (9)$$

where
*p(class)* = probability value of each class

Looking for the final result of each class is:

$p(attribute\ 1|class) * \ldots p(attribute\ n|class)$ = *value of each attribute based on class*

Internet of Things (IoT) is a new technology that can improve its users' quality of life and facilitate access to information and services provided. NodeMCU ESP8266 is an open source IoT platform and development kit or microcontroller to assist programmers in making prototypes of IoT products. NodeMCU is unique, where the size of the board is small with a length of 4.83 cm, a width of 2.54 cm and a weight of 7 grams. However, despite the small size of NodeMCU, this board is equipped with WiFi features and open source firmware. NodeMCU has advantages in terms of cost and space saving, because NodeMCU's size is small and practical. NodeMCU's price is much cheaper than Arduino Uno. It has a WiFi module, compared to Arduino Uno which does not yet have a WiFi module and is also slightly larger in size.

An HC-SR04 ultrasonic sensor is a 40 KHz sensor used to measure distance between an object and the sensor. Pins on the ultrasonic sensor includes:

1. Trigger
   This pin is the input pin of the sensor to generate ultrasonic waves.
2. Echo
   This pin is the output pin of the sensor to detect the reflection of ultrasonic waves.
3. VCC
   This pin is to power the sensor with +5V power.
4. GND
   Ground/0V power supply as a negative voltage source.

This sensor has 2 main components:

1. Ultrasonic transmitter
   Emits ultrasonic waves with a frequency of 40 KHz.
2. Ultrasonic receiver
   Captures the reflection of ultrasonic waves that hit an object.

To start a measurement, the trigger pin must be made high for 10 μs then turned off to trigger ultrasonic waves at a frequency of 40 Hz from the transmitter/ultrasonic transmitter. The receiver/ultrasonic receiver will wait for the waves to return. The distance between the sensor and the object can be calculated using the following equation:

$$s = \frac{340 * t}{2} \quad (11)$$

where
$s$ = distance.
$t$ = the time difference between the emitted and received waves.

An MQ-4 sensor is a sensor used to detect methane gas and outputs its reading as an analog or digital voltage. This sensor has a high sensitivity to natural gas. It can operate at -10 to 50 °C. The pins of the MQ-4 sensor are:

1. Digital out (DO)
   This pin can also be used to get the output value by using a potentiometer.
2. Analog Out (AO)
   This pin is used to get 0-5V analog voltage based on gas intensity.
3. VCC
   This pin is for the power pin. The operating voltage is +5V.
4. GND
   This pin is to connect the sensor to the ground.

The overall monitoring system is illustrated in Fig. 1. The monitoring process starts from the sensor device that collects trash height data using the HC-SR04 ultrasonic sensor and the methane gas data using the MQ-4 sensor.

The power source for these sensors comes from a power bank connected to NodeMCU ESP8266. The data from the sensors are received and managed by NodeMCU. After the data are obtained by the microcontroller, the data will be sent to the delivery address via the Wi-Fi module, stored in the database, and displayed in the web application.

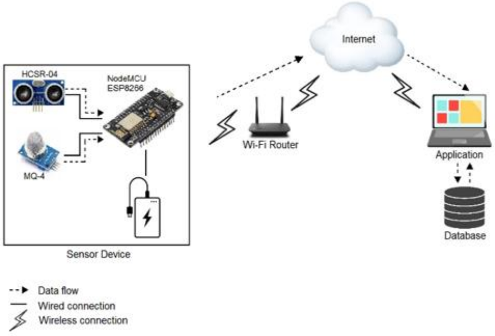


Figure 1. The trash can monitoring system

As shown in Fig. 2, a sensor device consists of an MQ-4 gas sensor to measure methane gas level that comes from the decomposition process of organic food, an HC-SR04 ultrasonic sensor to measure physical distance from the trash to the sensor, one NodeMCU ESP8266 that functions as a data receiver and sender, and a power bank to provide power to the sensor device.

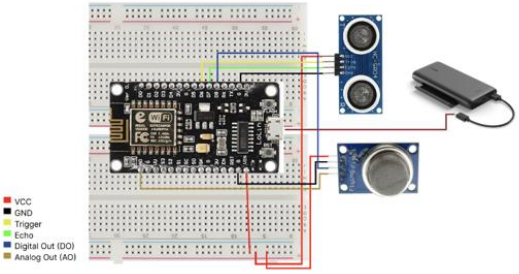


Figure 2. The design of a sensor device

The pins configuration between NodeMCU and the two sensors is given in Table 1. For the ultrasonic sensor, we use 4 jumper cables to connect the sensor pins to the NodeMCU pins. The GND pin functions as negative current in the circuit, the VCC pin draws 5V power from NodeMCU, the trig pin triggers the ultrasonic wave transmitter, and the echo pin detects the ultrasonic wave that bounces back and also transmits data from the sensor to NodeMCU. For the MQ-4 sensor, there are also 4 jumper cables used to connect the sensor pins to the NodeMCU pins. The GND pin functions as negative current in the circuit, the VCC pin draws 5V power from NodeMCU, the digital out (DO) pin provides numbers for the odor level from 0 to 500, and the analog out (AO) pin provides numbers for the odor level in the form of 1 and 0.

Table 1. Pins configuration

| NodeMCU | HC-SR04 | MQ-4 |
|---|---|---|
| D6 | Trig | - |
| D7 | Echo | - |
| A0 | - | Analog Out (AO) |
| A8 | - | Digital Out (DO) |
| VCC | VCC | VCC |
| GND | GND | GND |

Fig. 3 shows the wiring implementation of the sensor device in a trash can.

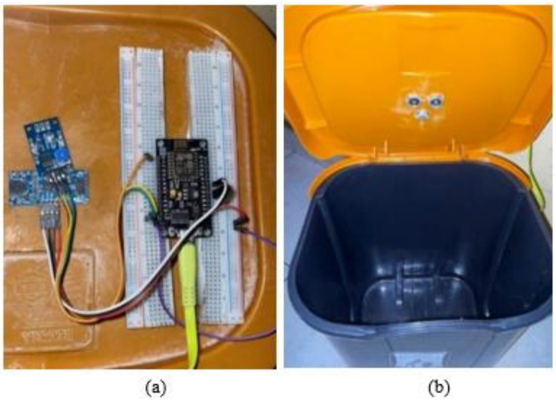

(a) (b)

Figure 3. (a) Top view and (b) inside look of the device implementation

Fig. 4(a) shows a flowchart for the client side (sensor device). The flow starts with device initialization. After that, the next step is to check if the Internet connection is available for communication between the device and the server (web application). If there is no connection, the device will be initialized and the connection will be checked again. If the connection has been established, the next step is to retrieve data from sensors and send them to the server. As long as the device is powered, it will keep retrieving and sending data to the server.

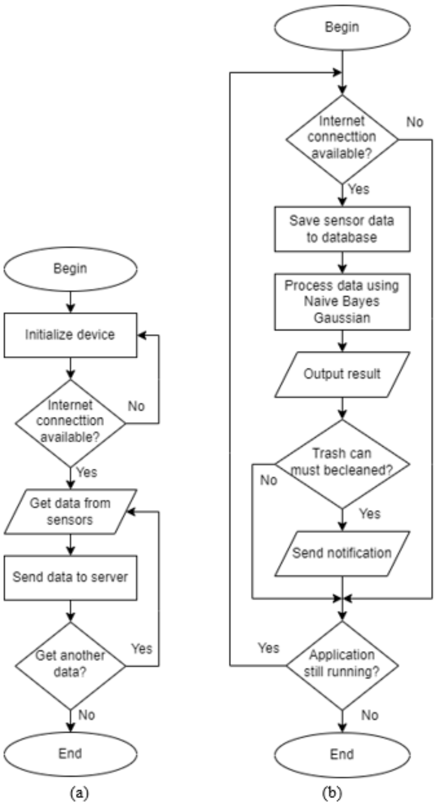


Figure 4. (a) Client flowchart and (b) server flowchart

Fig. 4(b) depicts a flowchart for the server side (web application). The procedure starts by checking whether the Internet is connected or not. If it is already connected, the sensor data from NodeMCU will be received by the application and will be stored in the database. Then, the data will be processed by using the Gaussian Naïve Bayes algorithm to obtain classification results that will be displayed on the user's screen. After the algorithm produces the classification results, if the trash can must be cleaned, the web application will send a notification to the user. This procedure repeats as long as the application is still running.

The preparation and steps for implementing the Gaussian Naïve Bayes algorithm are as follows.

### *3.1. Determine class indicators*

Table 2. Class indicators

| Capacity | Smell | Result |
|---|---|---|
| Quite Empty | | |
| >= 20 cm | <= 348 | Normal |
| >= 20 cm | 349 – 552 | Need To Be Cleaned |
| >= 20 cm | >= 553 | Really Need To Be Cleaned |
| Half Full | | |
| 19 – 11 cm | <= 348 | Normal |
| 19 – 11 cm | 349 – 552 | Need To Be Cleaned |
| 19 – 11 cm | >= 553 | Really Need To Be Cleaned |
| Almost Full | | |
| 10 – 6 cm | <= 348 | Need To Be Cleaned |
| 10 – 6 cm | 349 – 552 | Need To Be Cleaned |
| 10 – 6 cm | >= 553 | Really Need To Be Cleaned |
| Full | | |
| <= 5 cm | <= 348 | Need To Be Cleaned |
| <= 5 cm | 349 – 552 | Really Need To Be Cleaned |
| <= 5 cm | >= 553 | Really Need To Be Cleaned |

The class indicators in Table 2 are the results of our observations that will also be the basis for determining labels for accuracy tests.

### *3.2. Data from trash*

Table 3. Dataset

| Ultrasonic Value | MQ-4 Value | Result |
|---|---|---|
| 9 | 385 | Really Need To Be Cleaned |
| 9 | 682 | Really Need To Be Cleaned |
| 9 | 578 | Really Need To Be Cleaned |
| 10 | 427 | Really Need To Be Cleaned |
| 10 | 405 | Really Need To Be Cleaned |
| 10 | 430 | Really Need To Be Cleaned |
| 11 | 430 | Need To Be Cleaned |
| 11 | 424 | Need To Be Cleaned |
| 11 | 248 | Need To Be Cleaned |
| 10 | 247 | Need To Be Cleaned |
| 10 | 242 | Need To Be Cleaned |
| 10 | 240 | Need To Be Cleaned |
| 10 | 236 | Need To Be Cleaned |
| 10 | 228 | Need To Be Cleaned |
| 28 | 244 | Normal |
| 20 | 253 | Normal |
| 12 | 236 | Normal |
| 11 | 229 | Normal |
| 13 | 234 | Normal |
| 14 | 228 | Normal |
| 17 | 236 | Normal |
| 15 | 233 | Normal |
| 15 | 236 | Normal |
| 15 | 236 | Normal |
| 23 | 233 | Normal |

Table 3 shows the training data which are the basis for searching the classification results in the Gaussian Naïve Bayes algorithm.

### *3.3. Gaussian naïve bayes measures*

- Finding probability value from the normal, need to be cleaned, and really need to be cleaned classes.

  $P(\textit{Normal}) = \frac{11}{25} = 0.44$

  $P(\textit{Need To Be Cleaned}) = \frac{8}{25} = 0.32$

  $P(\textit{Really Need To Be Cleaned}) = \frac{6}{25} = 0.24$

- Finding mean value ($\bar{x}$) and variance ($\sigma$) from each attribute based on class.

  <u>Normal class:</u>

  $\textit{Mean}(\textit{Ultrasonic}) = \frac{(28+20+12+13+\cdots+23)}{11} = \frac{183}{11}$

  $= 16.63$

  $\textit{Variance}(\textit{Ultrasonic}) = \frac{(28-16.63)^2+\cdots+(23-16.63)^2}{11-1}$

  $= 26.25$

  $\textit{Mean}(\textit{MQ-4}) = \frac{(244+253+236+229+\cdots+233)}{11}$

  $= \frac{2598}{11} = 236.18$

  $\textit{Variance}(\textit{MQ-4}) = \frac{(244-236.18)^2+\cdots+(233-236.18)^2}{11-1}$

  $= 48.76$

  <u>Need to be cleaned class:</u>

  $\textit{Mean}(\textit{Ultrasonic}) = \frac{(11+11+11+10+\cdots+10)}{8} = \frac{83}{8}$

  $= 10.37$

  $\textit{Variance}(\textit{Ultrasonic}) = \frac{(11-10.37)^2+\cdots+(10-10.37)^2}{8-1}$

  $= 0.26$

  $\textit{Mean}(\textit{MQ-4}) = \frac{(430+424+248+247+\cdots+228)}{8} = \frac{2295}{8}$

  $= 286.87$

  $\textit{Variance}(\textit{MQ-4}) = \frac{(430-286.87)^2+\cdots+(228-286.87)^2}{8-1}$

  $= 7522.125$

  <u>Really need to be cleaned class:</u>

  $\textit{Mean}(\textit{Ultrasonic}) = \frac{(9+9+9+10+\cdots+10)}{6} = \frac{57}{6} = 9.5$

  $\textit{Variance}(\textit{Ultrasonic}) = \frac{(9-9.5)^2+\cdots+(10-9.5)^2}{6-1} = 0.3$

  $\textit{Mean}(\textit{MQ-4}) = \frac{(385+682+578+427+\cdots+430)}{6} = \frac{2907}{6}$

  $= 484.5$

  $\textit{Variance}(\textit{MQ-4}) = \frac{(385-484.5)^2+\cdots+(430-484.5)^2}{6-1}$

  $= 14049.1$

- Finding probability from new data based on class.
- If there is a new data, where ultrasonic = 10 and MQ-4 = 300.

$$P(10|Normal) = \frac{1}{26.25\sqrt{2*3.14}} e^{\frac{-(10-16.63)^2}{2*26.25^2}}$$
$$= 0.015e^{-0.031}$$

$$P(300|Normal) = \frac{1}{48.76\sqrt{2*3.14}} e^{\frac{-(300-236.18)^2}{2*48.76^2}}$$
$$= 0.008e^{-0.856}$$

$$P(10|Need\ To\ Be\ Cleaned)$$
$$= \frac{1}{0.26\sqrt{2*3.14}} e^{\frac{-(10-10.37)^2}{2*0.26^2}} = 1.48e^{-0.98}$$

$$P(300|Need\ To\ Be\ Cleaned)$$
$$= \frac{1}{7522.125\sqrt{2*3.14}} e^{\frac{-(300-286.87)^2}{2*7522.125^2}} = 5.30e^{-1.52}$$

$$P(10|Really\ Need\ To\ Be\ Cleaned)$$
$$= \frac{1}{0.3\sqrt{2*3.14}} e^{\frac{-(10-9.3)^2}{2*0.3^2}} = 1.33e^{-1.388}$$

$$P(300|Really\ Need\ To\ Be\ Cleaned)$$
$$= \frac{1}{14049.1\sqrt{2*3.14}} e^{\frac{-(300-484.5)^2}{2*14049.1^2}} = 2.840e^{-8.623}$$

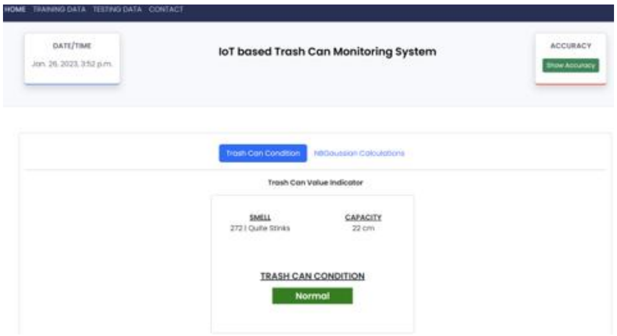
The result is:

$$Posterior(Normal) = 0.44 * 0.015e^{-0.031} * 0.008e^{-0.856} = 0.0000528e^{-0.887}$$

$$Posterior(Need\ To\ Be\ Cleaned) = 0.32 * 1.48e^{-0.98} * 5.30e^{-1.52} = 2.510e^{-2.5}$$

$$Posterior(Really\ Need\ To\ Be\ Cleaned) = 0.24 * 1.33e^{-1.388} * 2.840e^{-8.623} = 0.906e^{-10}$$

The largest value is *Posterior* (*Need To Be Cleaned*). Therefore, if the height of the trash is 10 cm and the methane gas level is 300, the trash can needs to be cleaned. Fig. 5 shows the trash can monitoring system's web application interface.



DATE/TIME
Jan. 26, 2023, 3:52 p.m.

IoT based Trash Can Monitoring System

Trash Can Value Indicator

SMELL
272 | Quite Stinks

CAPACITY
22 cm

TRASH CAN CONDITION
Normal

Figure 5. Trash can monitoring result using the Gaussian Naïve Bayes algorithm

## 4. Results and Discussion

To evaluate the system, we first check if each sensor functions correctly. For testing the HC-SR04 ultrasonic sensor, we compare the results of sensor measurements with the results of measurements using a manual measuring instrument. We check the results when the trash can is empty as shown in Fig. 6(a) and when the trash can is filled as shown in Fig. 6(b). The ultrasonic sensor gives the same measurement results as the measuring instrument.

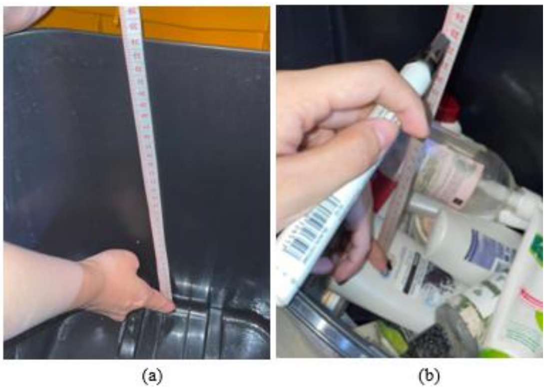


Figure 6. Testing the ultrasonic sensor in (a) an empty trash can and (b) a filled trash can

Figure 7. Testing the MQ-4 sensor in a trash can with (a) inorganic trash and (b) organic trash

For checking the MQ-4 gas sensor, we filled the trash can with two types of trash – inorganic and organic – as shown in Fig. 7. The MQ-4 gives different readings for these two conditions. For the inorganic trash in Fig. 7(a), the methane gas reading is 183, while for the organic trash in Fig. 7(b), the methane gas reading is 535. In Fig. 7(b), some organic trash has started to decompose and releases methane gas that produces bad odor.

To enhance the reproducibility of the study, the data collection process was conducted within a controlled urban environment to ensure a diverse range of waste disposal scenarios. Sensor nodes, equipped with ultrasonic and gas sensors, were programmed to sample environmental data at 30-minute intervals, capturing variables such as ultrasonic and MQ-4 values. This resulted in a raw dataset comprising 420 individual data points. Each data entry was logged via Wi-Fi to a centralized database, which served as the foundation for subsequent model training.

To establish a reliable ground truth for the supervised learning model, a systematic labeling protocol was implemented based on predefined physical thresholds as summarized in Table 2. The dataset was categorized into three distinct classes – normal, need to be cleaned, and really need to be cleaned – determined primarily by the distance measured between the ultrasonic sensor and the waste surface and the gas sensor thresholds.

Table 4. Accuracy testing of Naïve Bayes

| Test No. | Accuracy (%) | | |
|---|---|---|---|
| | 60:40 | 70:30 | 80:20 |
| 1 | 80.8 | 81.4 | 82.6 |
| 2 | 74.4 | 76 | 83.7 |
| 3 | 80.2 | 85.3 | 79.1 |
| 4 | 80.8 | 80.6 | 75.6 |
| 5 | 76.2 | 75.2 | 81.4 |
| 6 | 75 | 82.2 | 69.8 |
| 7 | 80.2 | 82.2 | 66.3 |
| 8 | 81.4 | 83.7 | 83.7 |
| 9 | 76.2 | 76.7 | 76.7 |
| 10 | 76.2 | 77.5 | 80.2 |
| 11 | 76.7 | 82.2 | 75.6 |
| 12 | 73.8 | 75.2 | 86 |
| 13 | 79.7 | 82.9 | 84.9 |
| 14 | 81.4 | 73.6 | 87.2 |
| 15 | 79.7 | 78.3 | 79.1 |
| 16 | 75.6 | 82.2 | 81.4 |
| 17 | 82 | 77.5 | 75.6 |
| 18 | 72.1 | 77.5 | 82.6 |
| 19 | 87.8 | 84.5 | 74.4 |
| 20 | 86.6 | 72.1 | 74.4 |
| 21 | 82 | 72.9 | 67.4 |
| 22 | 78.5 | 77.5 | 82.6 |
| 23 | 83.7 | 78.3 | 82.6 |
| 24 | 83.1 | 82.2 | 74.4 |
| 25 | 76.7 | 80.6 | 80.2 |
| 26 | 80.2 | 78.3 | 82.6 |
| 27 | 82 | 86 | 82.6 |
| 28 | 75.6 | 69.8 | 82.6 |
| 29 | 86 | 81.4 | 81.4 |
| 30 | 79.1 | 80.6 | 75.6 |
| 31 | 80.2 | 83.7 | 82.6 |
| 32 | 78.5 | 79.1 | 79.1 |
| 33 | 82.6 | 86 | 73.3 |
| 34 | 83.1 | 76 | 74.4 |
| 35 | 72.1 | 80.6 | 77.9 |
| 36 | 81.4 | 83.7 | 81.4 |
| 37 | 75 | 78.3 | 72.1 |
| 38 | 77.9 | 75.2 | 76.7 |
| 39 | 80.8 | 81.4 | 80.2 |
| 40 | 79.7 | 74.4 | 72.1 |
| 41 | 74.4 | 79.8 | 77.9 |
| 42 | 72.7 | 82.2 | 84.9 |
| 43 | 81.4 | 73.6 | 76.7 |
| 44 | 82.6 | 73.6 | 82.6 |
| 45 | 79.7 | 79.8 | 86 |
| 46 | 74.4 | 72.1 | 82.6 |
| 47 | 75.6 | 79.1 | 75.6 |
| 48 | 75 | 75.2 | 80.2 |
| 49 | 76.2 | 79.8 | 80.2 |
| 50 | 83.1 | 77.5 | 87.2 |
| Average | 79% | 78.9% | 79.1% |

We evaluate the system by testing the performance of the Naïve Bayes algorithm. The 420 collected data are divided into training data and testing data using 60:40, 70:30, and 80:20 split data. The data attributes are capacity (ultrasonic value) and smell (MQ-4 value), while the condition attribute is the classification result. The training data are used as benchmark for the testing data to classify the condition of the trash can. Table 4 shows the system's accuracy testing averaged over 50 runs, where data are split randomly. In the accuracy testing, the 60:40 partition's average result is 79%, the 70:30 partition's average result is 78.9%, and the 80:20 partition's average result is 79.1%. These findings demonstrate that the Naïve Bayes algorithm is robust across various train-test ratios, indicating high model stability.

The achievement of ~80% accuracy demonstrates that the Gaussian Naive Bayes model is sufficiently reliable in distinguishing waste conditions – specifically identifying organic waste beginning to decay based on gas concentration – while maintaining a light computational load. By implementing this model, sanitation workers can receive high-priority notifications exclusively for bins labeled 'Really Need To Be Cleaned'. This approach optimizes collection routes and significantly reduces operational time compared to conventional, inefficient scheduled cleaning systems. However, while Naive Bayes is exceptionally efficient for IoT devices, the ~80% accuracy rate implies a minor margin for error, suggesting that false alarms may still occasionally occur during real-time monitoring.

We further evaluate our system by comparing the Gaussian Naive Bayes algorithm to four other well-known algorithms: Random Forest, K-Nearest Neighbor (K-NN), Logistic Regression, and Linear Regression. The comparisons of accuracy, training time, prediction time, and memory consumption are shown in Figs. 8, 9, 10, and 11, respectively. All simulations utilize the 420 collected data points, which are divided using 60:40, 70:30, and 80:20 split ratios. The results presented are the average values from 50 runs.

Fig. 8 illustrates that while Gaussian Naive Bayes outperforms Linear Regression, it yields slightly lower accuracy than Logistic Regression.

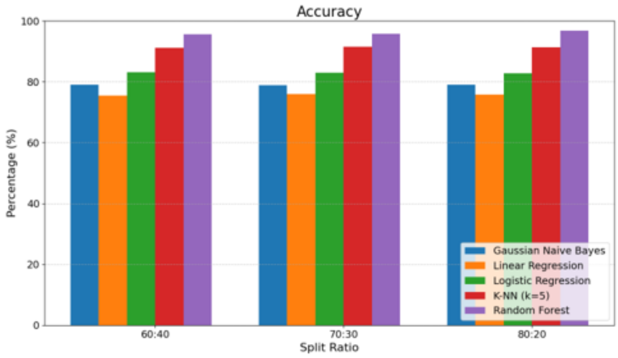


Figure 8. Accuracy comparison

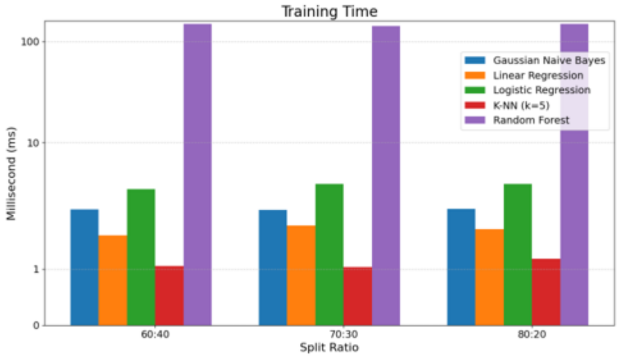


Figure 9. Training time comparison

However, Logistic Regression requires higher training time (Fig. 9) and memory consumption (Fig. 11). The highest accuracy is achieved by Random Forest, albeit at the cost of slower training and prediction times, as well as higher memory consumption—making it less suitable for IoT applications with strict battery and memory constraints. Furthermore, while the lazy learner K-NN offers a faster training time, it suffers from a slow prediction phase and higher memory usage compared to Gaussian Naive Bayes. Consequently, for this specific IoT application, Gaussian Naive Bayes effectively trades off a degree of accuracy (~80%) for superior computational efficiency.

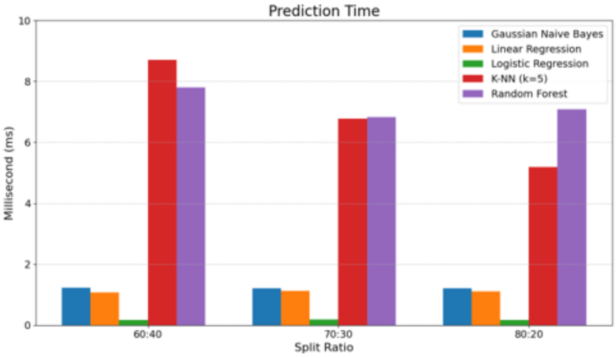


Figure 10. Prediction time comparison

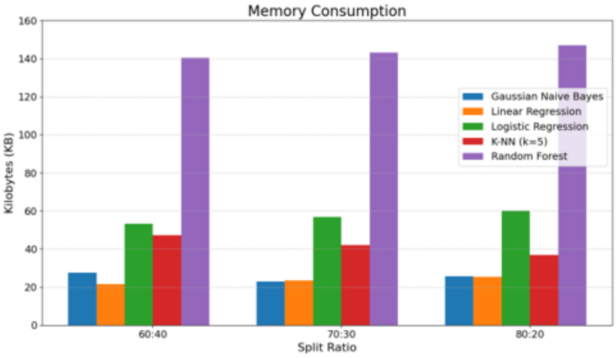


Figure 11. Memory comparison system

## 5. Conclusion

This research develops a resource-efficient, IoT-enabled trash can monitoring system to show that intelligent waste management can be effectively localized on low-power hardware like the NodeMCU. While more complex machine learning models often demand significant computational resources, the implementation of the Gaussian Naïve Bayes algorithm effectively bridges the gap between high-level predictive analytics and the constraints of distributed, memory-limited IoT nodes. The integration of ultrasonic and gas sensors provided a robust framework for monitoring both waste volume and odor, with experimental results yielding a reliable ~80% accuracy rate alongside minimal processing latency and memory consumption. Despite these promising results, the study has certain limitations. The current model's accuracy, while efficient for low-power nodes, still trails behind more resource-intensive algorithms like Random Forest or K-NN, which may be necessary for higher-stakes industrial applications.

## References

[1] A. Tanam, E. Keerthana, P. Ruchitha, and M. Tejasrikaveri, "IoT Enabled Garbage Maintenance for Railway Hygiene System," in *IEEE 6th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA)*, Hamburg: IEEE, 2024, pp. 307–312. doi: 10.1109/ICCCMLA63077.2024.10871683.

[2] S. N. Omer *et al.*, "Artificial Intelligence and Machine Learning Techniques in Solid Waste Management: A Sustainable Way Toward Future," *Comput. Sci. Rev.*, vol. 60, p. 100889, 2026, doi: 10.1016/j.cosrev.2025.100889.

[3] S. Adeoye, "Smart Waste Management Systems: An IoT and AI-Driven Approach for Urban Sanitation," *Comput. Inf. Sci.*, vol. 18, no. 2, pp. 30–49, 2025, doi: 10.5539/cis.v18n2p30.

[4] R. Gould, "Architecting a Connected Future," Geneva, 2019.

[5] J. B. Sanger, L. Sitanayah, and I. Ahmad, "A Sensor-based Garbage Gas Detection System," in *IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, NV: IEEE, 2021, pp. 1347–1353. doi: 10.1109/CCWC51732.2021.9376147.

[6] I. Rafiq, A. Mahmood, S. Razzaq, S. H. M. Jafri, and I. Aziz, "IoT Applications and Challenges in Smart Cities and Services," *J. Eng.*, vol. 2023, no. 4, p. e12262, 2023, doi: 10.1049/tje2.12262.

[7] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, "IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8182–8201, 2019, doi: 10.1109/JIOT.2019.2935189.

[8] S. Mehta and A. Singh, "Real-Time Data and Waste Management Efficiency: Insights from IoT-Enabled Smart Bins," in *International Conference Electrical Energy Systems*, 2024, pp. 1–5. doi: 10.1109/ICEES61253.2024.10776870.

[9] N. Acharya, R. P. Kewat, M. Kaif, R. Hossen, and M. Kumaresan, "EcoSense: A GSM-Enabled IoT Solution for Intelligent Waste Management," in *5th International Conference for Emerging Technology (INCET)*, Belgaum: IEEE, 2024, pp. 1–4. doi: 10.1109/INCET61516.2024.10593240.

[10] P. Rathnayake, T. M. M. Chanaka, P. T. R. Dabare, and K. P. Hewagamage, "IoT-Enabled Dual-Sensing Smart Waste Management System: Enhancing Urban Cleanliness and Sustainability in Smart Cities," in *8th SLAAI International Conference on Artificial Intelligence (SLAAI-ICAI)*, Ratmalana: IEEE, 2024, pp. 1–6. doi: 10.1109/SLAAI-ICAI63667.2024.10844960.

[11] K. Srividya, J. Nivedhana, R. Amrudha, and M. Vijayaraj, "Effective Sentitrash System for Urban and Rural Environments using IoT and ML Technology- : A Comprehensive Solution Integrating Multiple Sensors with ESP32 and Arduino Platforms," in *International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*, Chennai: IEEE, 2024, pp. 1–6. doi: 10.1109/ICPECTS62210.2024.10780305.

[12] Y. A, S. N, M. K, M. Y. B, and T. S, "IoT-based Smart Garbage Monitoring and Alert System to Track the Dustbin in Smart Cities," in *International Conference on Electronics and Renewable Systems (ICEARS)*, Tuticorin: IEEE, 2026, pp. 295–301. doi: 10.1109/ICEARS67481.2026.11416727.

[13] V. Tundjungsari, B. P. D. Putranto, M. B. Ulum, and N. Anwar, "An Integrated Model for Circular Waste Management Using the Internet of Things, Semantic Web, and Gamification (Circonomy): Case Study in Indonesia," *JMIR Serious Games*, vol. 13, 2025, doi: 10.2196/66781.

[14] S. Pulparambil, A. Al-Busaidi, Y. Al-Hatimy, and A. Al-Farsi, “Internet of Things-based Smart Medical Waste Management System,” *Telemat. Informatics Reports*, vol. 15, p. 100161, 2024, doi: 10.1016/j.teler.2024.100161.

[15] L. C. Thirumalai, R. Meenakshi, C. Suma, E. Sivanantham, N. Arunfred, and S. V E, “IoT-Enabled Food Waste Upcycling and Redistribution using Naive Bayes,” in *International Conference on Innovative Data Communication Technologies and Application (ICIDCA)*, Coimbatore: IEEE, 2025, pp. 558–563. doi: 10.1109/ICIDCA66325.2025.11280414.

[16] M. A. Febriantono, A. Pramono, and F. I. Maulana, “Developing Smart Trash Management System using IoT and Machine Learning,” in *International Conference on Information Technology Research and Innovation (ICITRI)*, Jakarta: IEEE, 2023, pp. 139–144. doi: 10.1109/ICITRI59340.2023.10250025.

[17] S. M., H. J., K. S. S. Kumar, S. P. S. Prakash, and M. J., “IoT-based Efficient Waste Management on University Campus : Enhancing Monitoring and Optimization through Machine Learning,” *Procedia Comput. Sci.*, vol. 258, pp. 4157–4168, 2025, doi: 10.1016/j.procs.2025.04.666.

[18] K. Hemachandran, S. Tayal, P. M. George, and U. Kose, *Bayesian Reasoning and Gaussian Processes for Machine Learning Applications*, 1st ed. Boca Raton: CRC Press, 2022. doi: 10.1201/9781003257491.