JOURNAL OF SOFT COMPUTING AND DATA MINING
e-ISSN: 2716-621X

JSCDM

Vol. 6 No. 3 (2025) 57-70
https://publisher.uthm.edu.my/ojs/index.php/jscdm

An Integrated Self-Diagnostic and Communication System for Autonomous Trash Can Robots

Bima Sena Bayu Dewantara[1]*, Kisron[1], Hary Oktavianto[2]

[1] *Department of Informatics and Computer Engineering, Politeknik Elektronika Negeri Surabaya, Surabaya, 60111, INDONESIA*
[2] *Department of Electrical Engineering, Politeknik Elektronika Negeri Surabaya, Surabaya, 60111, INDONESIA*

*Corresponding Author: bima@pens.ac.id
DOI: https://doi.org/10.30880/jscdm.2025.06.03.005

Article Info

Received: 26 April 2025
Accepted: 19 May 2025
Available online: 30 December 2025

Keywords

Trash bin robot, self-diagnostic, IoT-based communication, Naive Bayes, MySQL database

Abstract

This paper introduces two interrelated supporting features of an intelligent trash can robot system: an AI (Artificial Intelligence) -driven self-diagnostic system and an IoT (Internet of Things) communication system for human-robot collaboration. The self-diagnostic system assesses the robot's operational status based on internal and external parameters. When critical conditions arise, such as a full trash can, low battery, or operational failure, the robot communicates with operator via a Telegram bot, providing real-time status updates. Experimental results demonstrate that the Naive Bayes-based self-diagnostic system achieves an accuracy of 0.97 and an F1 score of 0.95. Furthermore, a developed communication system based on MySQL database significantly reduces data transmission time from 5 seconds using Wemos D1 mini to under 2 seconds.

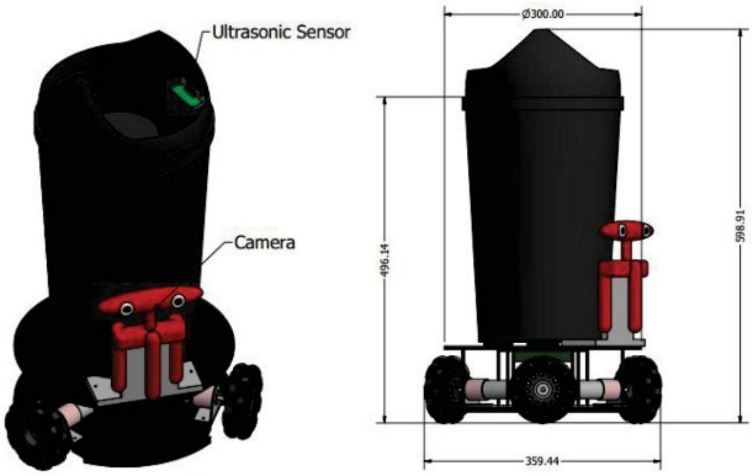
1. Introduction

The escalating demands of urban living and the growing emphasis on sustainable practices have catalyzed significant interest in developing and deploying autonomous robots designed for environmental management [1]. The World Bank estimates that every year, urban areas produce 2.01 billion tonnes of solid waste, of which at least 33% is not managed in an environmentally safe way [2]. Up to 70% of municipal waste management budgets in densely populated areas are allocated to manual waste collection, which results in inefficiencies and expensive operating expenses [3]. Furthermore, research indicates that postponing garbage collection can raise exposure to hazardous waste and encourage the spread of disease vectors, which can lead to increased risks to public health and urban hygiene [4]. Automating waste management processes through autonomous trash collection robots offers a promising avenue for enhancing efficiency, reducing operational costs, and improving overall urban cleanliness. These robots can navigate complex environments, identify and collect various types of waste, and adapt to changing conditions in real-time [5].

The development of autonomous mobile robots necessitates robust self-diagnostic and communication systems to ensure efficient and reliable operation in dynamic environments [6]. Robots designed to navigate and collect trash autonomously require the ability to monitor their internal states, detect potential issues, and communicate effectively with a central control system, the officer, and other robots [7]. Integrating self-diagnostic capabilities allows the robot to identify potential technical problems that can be found early and remedied before total failure. The robot can also periodically check its own states (i.e. stuck, rollback, etc.) and components (in these cases the battery and the trash bin state) and change its working behavior or even ask for help if needed. [8].

This proactive approach enables timely maintenance and prevents unexpected breakdowns, minimizing unplanned downtime, can extend the life of the device, and maximizing operational efficiency [9]. Also, with self-diagnostics the robot can also replace many of the typical manual inspections done on a regular basis by technicians, and free up technicians' time and save labor costs, especially in 24/7 continual systems or where there are many devices in operational mode. Furthermore, a sophisticated communication system facilitates seamless data exchange between the robot and its operators, providing real-time status updates, diagnostic information, and environmental data [10]. The communication system can leverage various wireless technologies, such as Wi-Fi, Bluetooth, or cellular networks, to ensure reliable connectivity in diverse environments.

In this context, this paper focuses on designing, implementing, and evaluating an autonomous trash collection robot equipped with a sophisticated self-diagnostic system for fault detection and a communication system for an alert system, aiming to provide a comprehensive solution for modern waste management challenges. Robot ATRACBOT as shown in Fig. 1, is an autonomous robot developed with AI to increase public awareness of the importance of throwing trash into the trash bin. AI includes real-time detection and classification of trash [11], detection of target people around the robot after finding trash [12], autonomous navigation and maneuvering [13], and the ability to self-monitor and communicate with officers (the person responsible for cleaning the trash) using the Internet of Things. This paper focuses on developing self-monitoring and communication systems as features in supporting robot operation.



(a) **(b)**

Fig. 1 *The design of ATRACBOT (Autonomous Trash Can Robot) (a) Sensor placements; (b) Robot's dimension*

By utilizing AI technology, the development of a self-diagnostic system begins by comparing several machine learning methods such as Support Vector Machine (SVM), Artificial Neural Network (ANN), k-Nearest Neighbors (k-NN), Logistic Regression (LR), Decision Tree (DT), and Naive Bayes (NB). A deeper evaluation was performed by employing a cross-validation model and a split train-test model to demonstrate the performance of the self-diagnostic system using several types of classifiers. All classifiers are also compared to determine the best method for subsequent use.

Developing an IoT-based communication system between the robot and the officer is also another main focus. We employed and compared the performances of a Wemos D1 Mini microcontroller [14] with a MySQL database as a node to send and receive data between the officer and the robot. The comparison aimed at finding the least noticeable communication latency between the two. The Wemos D1 Mini microcontroller is connected to the sensor via the mini-PC serial port. It is used as a real-time data sending node so that when the officer asks the robot for data through the microcontroller, the microcontroller will reply with the latest data. On the other hand, to fulfill the same purpose, employing a MySQL database as sensor data storage which periodically receives data from the sensor. This database allows for faster data transmission because there is no need to communicate with sensor nodes. However, the sensor data might not be the newest.

Thus, we made the following contributions: (1) built a robust and high-accuracy self-diagnostic system to support robot operations specifically in working together with humans to collect trash and (2) developed a fast communication system between robots and officers using Internet of Things technology. As a result, self-diagnostic methods and communication systems will be developed to support the operation of ATRACBOT.

The rest of this paper is organized as follows: Section 2 describes other researchers' works related to self-diagnostic and communication systems. Section 3 describes our self-diagnostic system by comparing some artificial intelligence methods and our approach to the communication system. The experimental results and discussion are presented in Section 4. Finally, Section 5 concludes our work and outlines our future.

2. Related Works

IoT-based waste management systems have been a focus of attention in the past ten years, with a variety of approaches ranging from simple monitoring to the adoption of artificial intelligence and autonomous robots. Omar et al. [15] created the Spatial Smart Waste Management System, combined with the Smart City Service Delivery Engine, to monitor fill levels of bins and ensure contractor compliance, but it lacks a system-level device self-diagnostic component. Rajput and Patil [16] conducted a survey of ICT and IoT-based waste management strategies, such as capacity sensors, RFID, and wireless sensor networks, but the paper is predominantly theoretical in nature with no direct utility. Hong [17] designed a Smart Garbage System for managing food waste using a Wireless Mesh Network with resilience to node failures.

Juwariyah et al. [18] applied low-cost real-time monitoring using Wemos and the Blynk IoT platform, though without fault detection automation. Alourani et al. [19] integrated VGG-19 CNN and IoT for high-accuracy waste separation with mobile notification in integration, though device diagnostics are yet implicit using data anomaly detection. Alabdali [20] presented an even more integrated system using IoT, AI, and blockchain for categorization, path optimization, and anomaly detection for data and devices, thereby one of the most integrated solutions. Throughout all of these papers, there is a seen shift towards more inclusive systems that address three most critical pillars: extensive coverage of applications, diagnostic capabilities of devices, and error-free communication systems. In this paper, the proposed system (ATRACBOT) is an autonomous robot for detection of waste, navigation, and collection via computer vision, operational condition self-checking mechanisms, and robust wireless communication via local database and implementation of Telegram for remote control and telemetry. Table 1 shows the comparison of the proposed system and the other approaches to show the differences among each other.

Table 1 *The comparison of the proposed system and the other approaches*

Study & Year	Application	Self-Diagnosis		Communication System		Information
		Description	Yes/No/ Partial	Description	Yes/No	
Omar et al. [15]	Smart city waste monitoring & contractor enforcement	Limited – monitors collection performance, no hardware health check	Partial	Smart wireless sensor network with Spatial Smart City Service Delivery Engine	Yes	Strong monitoring & comms but lacks robust self-diagnostics
Rajput & Patil [16]	Survey of IoT waste management (collection, transport, recycling)	Potential – suggests sensor-based fault detection	Partial	IoT platform with WSN, RFID	Yes	Broad conceptual scope; No real implementation; effectiveness not tested
Hong [17]	Smart Garbage System for food waste	Limited – network dropout may indicate faults	Partial	Wireless Mesh Network (WMN)	Yes	Resilient network but limited diagnostic capability
Juwariyah et al [18]	Basic bin monitoring with Blynk IoT app	None – manual fault observation	No	Wi-Fi from Wemos to Blynk cloud	Yes	Very basic system; missing automated diagnostics
Alourani [19]	AI classification + smart collection (VGG-19 CNN)	Potential – AI can detect missing/erroneous data	Partial	IoT bins linked to ThingSpeak + Android alerts	Yes	Strong application; potential diagnostics; good comms

Alabdali [20]	IoT + AI + Blockchain waste classification & routing	Advanced – blockchain anomaly detection for sensors & data	Yes	IoT + blockchain-secured comms	Yes	Advanced application; robust self-diagnosis; secure comms
Ours (ATRACBOT)	Autonomous trash collection robot with AI vision & navigation	Strong – self-check for conditions based on sensors	Yes	MySQL database + Telegram Bot	Yes	Autonomous operation; operational condition self-checks; strong comms

3. Methods

The designed components of the robot related to self-diagnostic and communication systems are shown on Fig. 2. To support these modules, several sensors are used, including an ultrasonic sensor, power sensors, and an IMU (Inertial Measurement Unit) sensor. The whole system is built using the ROS (Robot Operating System) platform, enabling all subsystems to multitask with a more even distribution of CPU cores than by using sequential programming [21][22]. This is expected to maximize the existing resources of robots.

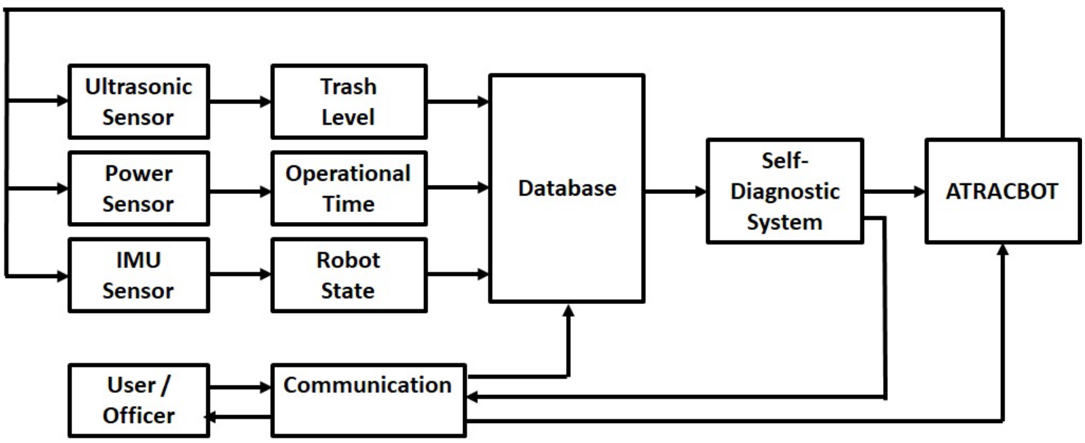


Fig. 2 *Main block diagram of the proposed system*

From Fig. 2, the self-diagnostic system is influenced by several parameters, namely the trash level measurement, operational time prediction, and the robot's state information. Ultrasonic sensors were employed to determine the level of trash in the robot. For the operational time prediction, a voltage sensor was used to read the state of charge of the battery to determine the remaining power. The Inertial Measurement Unit (IMU) is used to inform the robot's state by utilizing the 3-D position of the robot to know whether the robot is in a normal state or experiencing interference. Each sensor's data is then stored in the MySQL database. The result of the self-diagnostic system will be used by the robot to determine whether it should stop or continue working. Apart from that, the output of the self-diagnosis system will also be sent to the communication module to be informed to the user or officer as the monitoring function. The user or officer can interrupt the robot or the database through the communication module as the control function.

3.1 Self-Diagnostic System

Since our robot is designed to work autonomously without direct supervision from an officer, it is necessary to build a self-diagnostic system on the ATRACBOT so that the robot can make decisions independently based on the conditions that occur. We designed the input consisting of four parameters, namely motor's battery working time, embedded system's battery working time, trash level, and robot's state.

Battery working time is considered because the robot works using a voltage source from the battery. If the voltage source is completely exhausted, the robot cannot work. In this paper, by assuming that the battery is in an ideal condition, we skip the factor of decreasing the quality of our battery. In addition, trash level is also an important factor since the trash bin on the robot can become full at any time, causing the robot to be unable to accommodate the trash and thus must stop working. Finally, the importance of the robot's state is to know whether it is falling, stuck, or hitting another object, which may prevent it from performing its task.

3.1.1 Estimating Battery Working Time

To estimate the battery working time, two voltage sensors were used. The first voltage sensor measures the battery usage for the embedded system, and the second one measures the battery usage for the motors. Before using the sensors to measure the battery level, the error measurement made by the sensor was evaluated. To ensure the sensors' feasibility, the sensor measurements were compared with a multimeter and obtained that the sensor measurement error is less than 0.5%. It proves that the sensors can precisely measure the battery voltage. In this experiment, two 3-cell LiPo batteries, namely 1,800 mAh for motors and 5,500 mAh for mini-PC were used. Table 2 shows the battery performance under loaded and unloaded states.

Table 2 *The battery performance under loaded and unloaded states*

Battery Capacity	Used for	Loaded/Unloaded	Time
1,800 mAh	Motors	No active	330 minutes (5 hours 30 minutes)
		Active constant speed	75 minutes (1 hour 15 minutes)
5,500 mAh	Mini-PC (including sensors and communication module)	Standby	540 minutes (9 hours)
		Running active communication with interval of every 5 minutes	486 minutes (8 hours 6 minutes)
		Running online camera capture	210 minutes (3 hours 30 minutes)

The estimation of battery working time is obtained from calculations using voltage drop rate and Kalman filter. The Kalman filter is used to track and smooth the battery state of charge trend that is sampled by the voltage sensor. The actual voltage can be formulated as follows:

$$V_t = V_{t-1} - \dot{V}.\Delta t, \tag{1}$$

where V_t is the voltage at time t and V_{t-1} is the voltage at time $t - 1$. $\dot{V}$ is the battery voltage drop rate and Δt is the sampling time. Applying the Kalman filter to track and smooth the motor's and embedded system's battery state of charge will update Eq. (1) as follows:

$$\begin{aligned} V_t^m &= V_{t-1}^m - \dot{V}^m \Delta t + \epsilon^{V^m}, \\ V_t^{es} &= V_{t-1}^{es} - V^{es} \Delta t + \varepsilon^{V^{es}}, \\ \dot{V}_t^m &= \dot{V}_{t-1}^m + \epsilon^{V^m}, \\ \dot{V}_t^{es} &= \dot{V}_{t-1}^{es} + \epsilon^{V^{es}} \end{aligned} \tag{2}$$

where the suffixes "m" and "es" denote motor and embedded system, respectively. ϵ^V is noise applied to the voltage measurement. The state can be formulated as a tuple $X = [V_t^m, V_t^{es}, \dot{V}_t, \dot{V}_t^{es}]$ and our Kalman filter model is expressed as $X = F_t X_{t-1} + \varepsilon$, where F_t denotes the state transition model and $\varepsilon = N(0, Q)$ denotes the process noise which is supposed to be derived from a zero mean multivariate normal distribution with covariance $Q = [0.5\ 0.5\ 0.1\ 0.1]$. To find the estimated operational time of the battery, the voltage drop rate can be calculated from Eq. (1) as follows:

$$\dot{V} = \frac{V_t - V_{t-1}}{\Delta t} \tag{3}$$

After getting the battery voltage drop rate, the battery operational time can be estimated by following equation:

$$Estimated\ time\ = \frac{V_t - V_{min}}{\dot{V}}, \tag{4}$$

where V_{min} is the minimum limit of the battery. It has a value of 11.1 Vdc.

3.1.2 Trash Level Measurement

The trash level is obtained from the measurement results of the SRF-05 sensor. As shown in Fig. 3, the sensor is mounted face down on the lid of the trash can and reads the distance of the pile of trash. The higher the pile of trash, the smaller the ultrasonic sensor value reading will be.

Fig. 3 *Placement of ultrasonic sensors for measuring the height of the trash heap*

3.1.3 Robot State Determination

The robot's state is determined by using measurements from an IMU sensor, which detects abnormal conditions. Abnormalities include the robot stopping after hitting an obstacle, indicated by no change in position over time (from accelerometer data), or rolling over, indicated by a change in orientation (from gyroscope data). Categorizing these abnormalities into safe and failed states is important. Other than that, the batteries condition is also taken part to make robot's operational abnormalities. Table 3 presents sample data used in this paper.

The dataset collected represents a minimal sample of the physical system, which, if fully analyzed, would be extremely time-consuming given the large number of combinations. The dataset selection was intuitive, assuming the collected data adequately represented only the most important combinations of conditions. Each combination was represented by at least several data samples. Backup data was also added to anticipate corrupted data or outliers.

Table 3 *Dataset*

Estimation of Motor Battery (minutes)	Estimation of Mini PC Battery (minutes)	Trash Level (cm)	System State	Decision
21.25	43.7	17	safe	run
28.8	1.15	8	fail	stop
68.03	9.65	18	fail	stop
65.53	39.88	37	fail	stop
35.42	60.97	14	fail	stop
51.93	50.2	21	safe	run
0.07	27.29	11	safe	stop
9.3	13.88	12	fail	stop
13.78	44.37	15	safe	run
47.56	67.66	28	fail	stop
52.59	9	14	fail	stop
18.89	59.93	34	fail	stop
47.8	23.94	40	safe	run

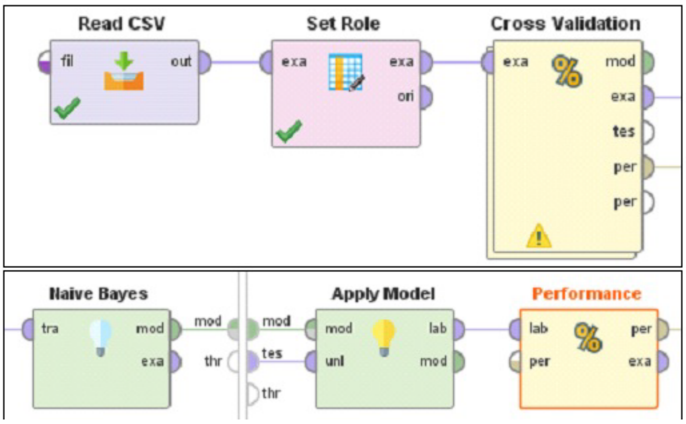
Estimation of Motor Battery (minutes)	Estimation of Mini PC Battery (minutes)	Trash Level (cm)	System State	Decision
17.4	8.67	21	safe	run
29.61	6.23	29	safe	run
78.11	42.85	22	fail	stop
9.25	76.4	11	safe	run
34.07	48.28	17	safe	run
36.81	65.91	35	fail	stop
26.96	58.94	2	safe	run
4.34	49.58	19	fail	stop
58.51	2.84	8	fail	stop
10.65	63.05	29	safe	run
32.85	70.45	12	safe	run
77.17	37.83	35	fail	stop
38.63	36.99	16	fail	stop
49.47	76.87	31	safe	run
12.1	78.15	37	safe	run
0.21	3.15	35	fail	stop
17.38	30.46	13	fail	stop

3.1.4 Artificial Intelligence-based Self-Diagnostic

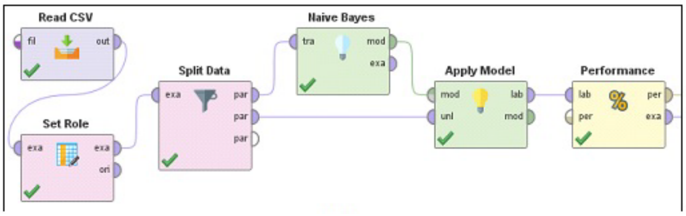
The self-diagnostic system is built using several classification methods, which are compared to selecting the best one. It outputs a decision on whether the robot should continue operating or stop, based on input data. In addition, operators can remotely request battery status, trash fullness level, estimated working time, and shut down the robot. This paper compares SVM, Neural Networks, k-Nearest Neighbors, Logistic Regression, Decision Trees, and Naive Bayes, using default parameters listed in Table 4.

Table 4 *Parameter*

Method	Parameter
Support Vector Machine	Kernel = RBF
	C = 0.0
	Gamma = 1.0
Neural Network Backpropagation	Hidden layer = 2
	Learning rate = 0.01
	Momentum = 0.9
	Training cycles = 200
k-Nearest Neighbors	K = 10
Logistic Regression	Reproducible = Yes
	Max number of thread = 4
	Missing value handling = MeanImputation
Decision Tree	Criterion = gain_ratio
	Max depth = 10
	Confidence = 0.1
	Minimal gain = 0.01
	Minimal leaf size = 2
Naive Bayes	Laplace correction = True



(a)



(b)

Fig. 4 *RapidMiner framework: (a) Cross-validation; (b) Split validation*

This paper uses a numerical dataset, which benefits the performance of all tested algorithms. Naive Bayes and k-NN are known for their high accuracy, with k-NN being particularly effective for low-dimensional datasets like the one used here [23]. Decision tree performance depends heavily on how well the tree is designed. Logistic regression tends to underfit unbalanced data, reducing accuracy [24]. Neural Networks, while powerful, require large datasets to perform well and did not outperform k-NN or Naive Bayes in this paper. Although SVM has shown strong results elsewhere [25], it was less effective here due to the lack of data normalization and its general-purpose nature [26], unlike Naive Bayes, which processes attribute individually [27].

RapidMiner software was used to compare machine learning methods, leveraging its flexibility in switching algorithms [28]. Experiments employed two evaluation methods: cross-validation and the split method. The k-fold cross-validation was applied, dividing the dataset into training and validation sets, as illustrated in Fig. 4(a) [29]. The train-test split method divides the existing dataset into two parts: training data and testing data. The training data is used to build classification models, while the testing data is used to test the models that have been constructed [30]. The train-test split method used in this paper uses 70% of data for training and 30% of data for testing. Figure 4(b) shows the framework of the train-test split.

3.2 Communication System

We developed a communication system using the Telegram application, which offers more features than other instant messaging platforms [31]. Fig. 5 illustrates two designs of the communication system using a Wemos D1 Mini microcontroller and a MySQL database. In Fig. 5(a), the microcontroller reads sensor data and sends it to a Mini PC for further processing. When an officer requests data, the Wemos D1 Mini receives the request, forwards it to the Mini PC, retrieves the latest sensor data via serial communication, and sends it back through the Wemos D1 Mini to the officer. While this ensures up-to-date data, the process is slower due to two layers of serial communication: between the sensor microcontroller and the Mini PC, and between the Wemos D1 Mini and the Mini PC.

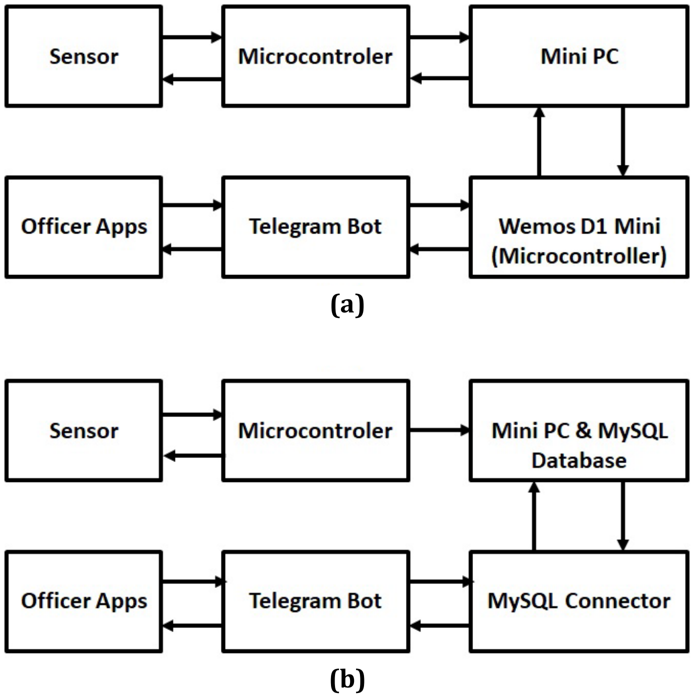


Fig. 5 *Difference in communication system design. (a) System design using Wemos D1 microcontroller; (b) System Design using MySQL database*

In Fig. 5(b), the communication system uses a database as temporary storage for sensor data, periodically sent from the microcontroller via the ROS framework [21][22]. A Telegram bot server, hosted on the robot's Mini PC connected to the internet, allows officers to access robot status remotely. The bot responds to user requests through the Telegram app. With ROS, CPU cores are efficiently distributed across nodes. Fig. 6 shows the system design using ROS, while Fig. 7 illustrates how the Telegram bot bridges communication between the user's app and the robot.

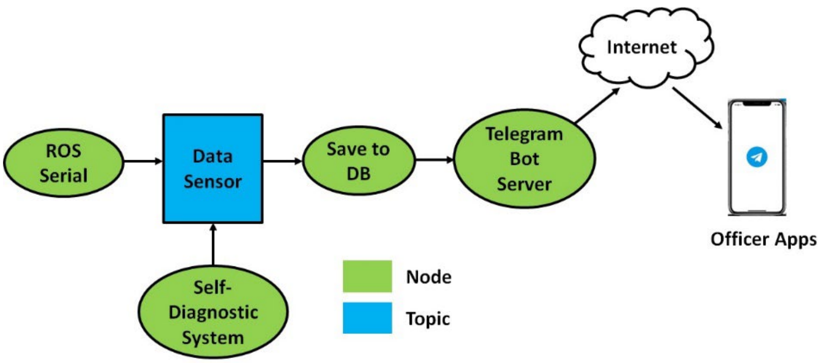


Fig. 6 *Communication system design*

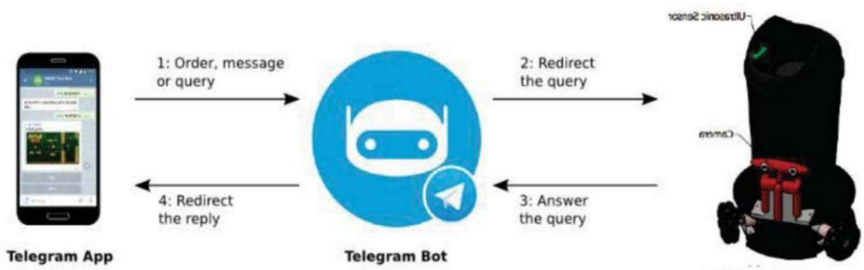


Fig. 7 *The TelegramBot work*

4. Results and Discussion

4.1 Self-Diagnosis System

In this section, the self-diagnostic system was tested using two models: 10-fold cross-validation and a train-test split. For cross-validation, 90% of the data (27 samples) was used for training and 10% (3 samples) for validation, with random splits repeated ten times. This method improves model performance by exposing it to different data variants [32]. Accuracy was calculated as the average across all folds. In the train-test split, 70% of the data (21 samples) was used for training and 30% (9 samples) for testing, with accuracy based on the percentage of correctly diagnosed cases.

Table 5 *Graph of performance comparison (CV: Cross validation; TTS: Train test split)*

Classification Method	Accuracy		Precision		Recall		F1 Score	
	CV	TTS	CV	TTS	CV	TTS	CV	TTS
Support Vector Machine	0.90	0.78	0.83	0.67	0.86	0.67	0.86	0.67
Neural Network	0.90	0.89	0.83	0.75	0.91	1.00	0.86	0.86
k-nearest neighbours	0.76	0.89	0.64	0.75	0.81	1.00	0.72	0.86
Linear Regression	0.90	0.89	0.83	1.00	0.91	0.67	0.86	0.80
Decision Tree	0.83	0.89	0.71	1.00	0.91	0.67	0.80	0.80
Naïve Bayes	0.97	0.89	1.00	1.00	0.91	0.67	0.95	0.80

The results, shown in Table 5, indicate that Naive Bayes outperformed other methods, achieving 0.97 accuracy and an F1 score of 0.95 in both tests. Its recall and F1 scores were also consistently strong, leading us to select Naive Bayes for ATRACBOT's self-diagnostic system. Naive Bayes was chosen for its combination of high accuracy, computational efficiency, and low memory usage. It also requires minimal training data, assuming attribute independence for fast and effective classification [33].

Discussing about ANN has better recall than Naive Bayes, this is because ANN is not bound by the feature independence assumption, can handle non-linear relationships, and is more liberal in avoiding false negatives. Naive Bayes will on the other hand be liberal and biased towards the majority class and hence have low recall. ANN can be tuned using special techniques (e.g., class weighting or oversampling) to better recognize minority classes. This makes ANN more proficient in recall than Naive Bayes, especially on complex and imbalanced data.

4.2 Communication System

In the experiment, a MySQL database was used as temporary storage for sensor data before sending it to officers via Telegram Messenger. Communication nodes were built using the Teleport library in Python. Fig. 8 compares data transmission times between the Wemos D1 Mini microcontroller and the MySQL database across 3G, 4G, and Wi-Fi connections. Results show significantly faster transmission when using the database. However, in Fig. 8, the previous 4G test results show higher latency than 3G due to the possibility of busy network conditions or a weaker 4G signal than 3G during the test. Therefore, the test results at that time did not show any advantage for 4G over 3G. Since the data was simple, a single table was sufficient (Fig. 9).

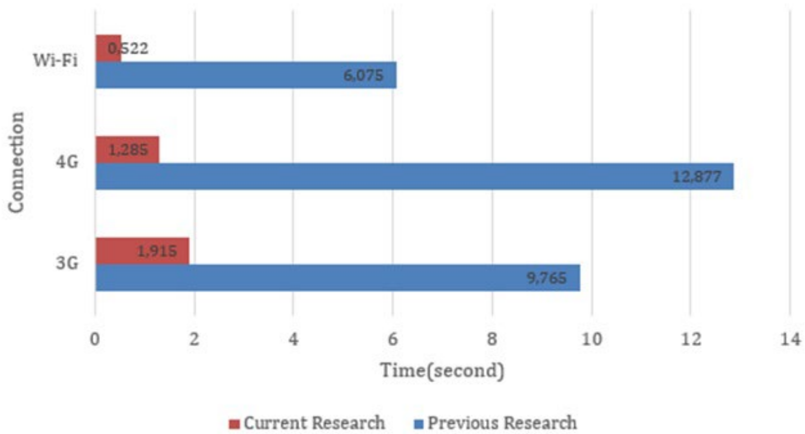


Fig. 8 *Comparison of data delivery times*

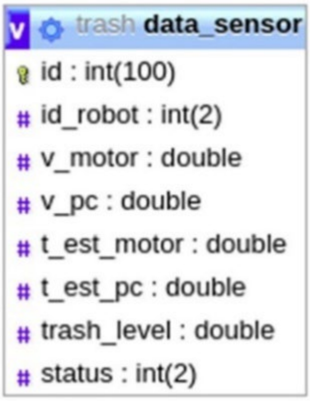


Fig. 9 *Table of sensor data*

Sensor data is stored in table columns, and indexing specifically on the "id" column was applied to speed up data retrieval [34]. When a user sends a command, the program fetches the latest entry by selecting the record with the highest "id". Communication with the database uses MySQL Connector, an ODBC-based driver [35]. Configuration includes the database host ("localhost"), username, password, and database name. A pseudo-code script retrieves the requested sensor data efficiently by accessing the latest value. Fig. 10 shows a mockup of our Telegram app, which not only displays messages but also offers commands to monitor and control the robot.

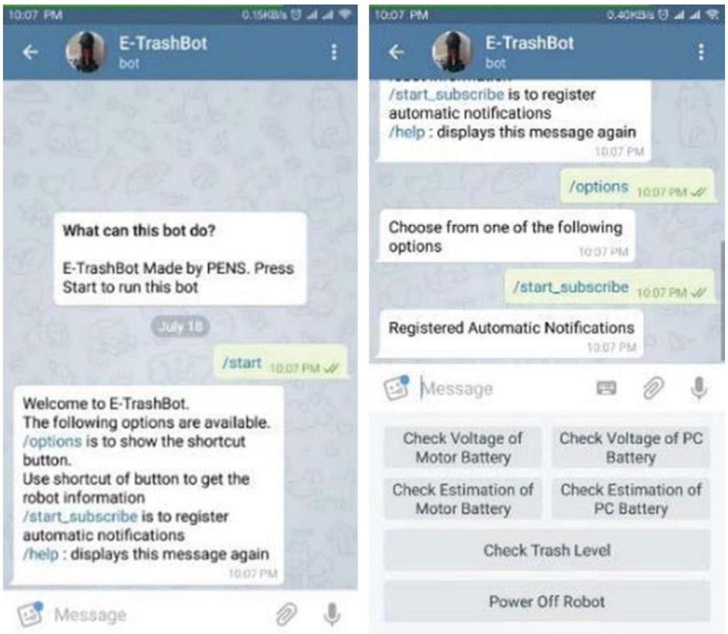


Fig. 10 *The display of our Telegram app for the user*

4.3 Energy Consumption

As one of the important parameters in supporting robot operations, the availability of energy by the battery is one of the important objects to evaluate. Every activity carried out by the robot, whether in the form of running the motor, accessing and processing sensor data, or communicating affects battery energy consumption. In Table 2, it is reported that when a 1,800 mAh battery is used to continuously provide energy to the motor, the battery capacity is reduced by up to 77.2% so that the operational time is reduced to 1 hour 15 minutes. Meanwhile, for the 5,500 mAh battery used to provide energy to the mini-PC, sensors, and communication module, it also experienced a significant reduction of 61.1% when running the camera application so that the operational time is reduced to 3 hours 30 minutes. When running the communication module to send data every 5 minutes, the battery capacity only reduced by 0.9%.

4.4 Behavior Analysis

The decision-making system was developed under the assumption that all sensor conditions work properly and ideally. However, during operation, system failures may occur due to errors or biases in sensor data readings, such as in the battery estimation system and state estimation system. These failures can lead to incorrect decision-making. In this section, the decision-making system is tested under two alternating conditions: a ±10% bias in battery estimation and a failure in state estimation, to simulate possible errors or biases in sensor data readings.

In the first simulation, which represents a state estimation failure, the estimated state was locked to a single condition, either “safe” only or “fail” only, while the battery estimation status was assumed to remain accurate. Each simulation test used 30 data variations. When the state estimation was locked to “safe” only, the decision-making system produced 12 incorrect decisions out of 30, resulting in an accuracy of 60%. When the state estimation was locked to “fail” only, the system produced 13 incorrect decisions out of 30, resulting in an accuracy of 56.57%.

In the second simulation, which represents a ±10% bias in battery estimation while the state estimation status was assumed to remain accurate, the results remained fully accurate at 100%. This indicates that the training dataset used is sufficiently representative of the possible conditions.

Based on these two tests, it is evident that bias in battery estimation does not significantly affect the robot’s decision-making. However, the robot’s state estimation has a highly significant influence on whether the robot continues to operate or stops.

5. Conclusion

We developed a self-diagnostic method by comparing several methods and developing a robot communication system with officers using MySQL database. The results of the testing with the evaluation method of 10-fold cross-validation showed that Naive Bayes achieved an accuracy of 0.97 and an F1 score of 0.95. Therefore, selecting Naive Bayes as a self-diagnostic method is entirely appropriate. In the communication system, the time required to send data using a local database and 3G connection was an average of 1.915 seconds, while with a 4G connection, the average time was 1.285 seconds. When using a Wi-Fi network, the average time required to send data was only 0.523 seconds. The use of a local database to temporarily hold data from a robot decreased the average time of sending data from initially more than 5 seconds to less than 2 seconds. Our next plan is to integrate the real system that has been made so that ATRACBOT can operate properly. We hope that ATRACBOT can increase public awareness of the importance of maintaining cleanliness and disposing of trash in its place.

Acknowledgement

The authors would like to thank the Center for Research and Community Services (P3M) of Politeknik Elektronika Negeri Surabaya (PENS) for funding this research. We also thank the members of Social Robotics and Smart System Applications (SRSSA) Research Group for helping our experiments.

Conflict of Interest

Authors declare that there is no conflict of interest regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Bima Sena Bayu Dewantara;* ***data collection:*** *Kisron;* ***analysis and interpretation of results:*** *Bima Sena Bayu Dewantara, Hary Oktavianto, Kisron;* ***draft manuscript preparation:*** *Bima Sena Bayu Dewantara, Kisron. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] Guenat, S., Purnell, P., Davies, Z. G., Nawrath, M., Stringer, L. C., Babu, G. R., and Dallimer, M. (2022). Meeting sustainable development goals via robotics and autonomous systems. *Nature communications, 13(1),* 3559. https://doi.org/10.1038/s41467-022-31150-5.

[2] Abdallah, M., Talib, M. A., Feroz, S., Nasir, Q., Abdalla, H., and Mahfood, B. (2020). Artificial intelligence applications in solid waste management: A systematic research review. *Waste Management, 109,* 231-246. https://doi.org/10.1016/j.wasman.2020.04.057.

[3] Nafiz, Md. S., Das, S. S., Morol, Md. K., Juabir, A. A., and Nandi, D. (2023). ConvoWaste: An Automatic Waste Segregation Machine Using Deep Learning. *In 2023 3rd International conference on robotics, electrical and signal processing techniques (ICREST),* 181-186. https://doi.org/ 10.1109/ICREST57604.2023.10070078.

[4] Vishnu, S., Ramson, S. R. J., Rukmini, M. S. S., and Abu-Mahfouz, A. M. (2022). Sensor-Based Solid Waste Handling Systems: A Survey. *Sensors, 22(6).* 2340. https://doi.org/10.3390/s22062340.

[5] Che, C., Zheng, H., Huang, Z., Jiang, W., and Liu, B. (2024). Intelligent robotic control system based on computer vision technology. *arXiv:2404*.01116. https://doi.org/10.48550/arXiv.2404.01116.

[6] He, H., Gray, J., Cangelosi, A., Meng, Q., McGinnity, T. M., and Mehnen, J. (2021). The challenges and opportunities of human-centered AI for trustworthy robots and autonomous systems. *IEEE Transactions on Cognitive and Developmental Systems, 14(4),* 1398-1412. https://doi.org/10.1109/TCDS.2021.3132282

[7] Torrejón, A., Zapata, N., Bonilla, L., Bustos, P., and Núñez, P. (2024). Design and Development of Shadow: A Cost-Effective Mobile Social Robot for Human-Following Applications. *Electronics 2024, 13, 3444. Advances in Human-Machine Interaction, Artificial Intelligence, and Robotics, 183*. https://doi.org/10.3390/electronics13173444.

[8] De Francesco, E., De Francesco, R., Leccese, F., De Francesco, E., and Petritoli, E. (2024, November). Preventive and Predictive Maintenance of Medium Complexity Autonomous Vehicles. *In 2024 IEEE International Workshop on Technologies for Defense and Security (TechDefense),* 221-225). IEEE. https://doi.org/10.1109/TechDefense63521.2024.10863247.

[9] Koul, N., Kumar, N., Sayeed, A., Verma, C., and Raboaca, M. S. (2022). Data exchange techniques for internet of robotic things: Recent developments. *IEEE Access, 10*, 102087-102106. https://doi.org/10.1109/ACCESS.2022.3209376.

[10] Salimi, I., Dewantara, B. S. B., and Wibowo, I. K. (2018, October). Visual-based trash detection and classification system for smart trash bin robot. *In 2018 international electronics symposium on knowledge creation and intelligent computing (IES-KCIC),* 378-383. IEEE. https://doi.org/10.1109/KCIC.2018.8628499.

[11] Dewantara, B. S. B., Ardilla, F., and Thoriqy, A. A. (2019) Implementation of depth-HOG based human upper body detection on a mini pc using a low-cost stereo camera. *2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT),* 458-463. http://doi.org/ 10.1109/ICAIIT.2019.8834580.

[12] Widarto, M. F. E., Marta, B. S., and Dewantara, B. S. B. (2023, August). Navigation System Development on ATRACBOT using F-SFM. *In 2023 International Electronics Symposium (IES*), 256-261. IEEE. https://doi.org/10.1109/IES59143.2023.10242566.

[13] Moris, F. D. and Widjaja, D. (2021, March). IOT based trash can monitoring system for smart garden cleanliness. *IOP Conference Series: Materials Science and Engineering, 1115(1), 012053.* IOP Publishing. http://doi.org/10.1088/1757-899X/1115/1/012053.

[14] Budiharto, W., Andreas, V., Irwansyah, E., Suroso, J. S., & Gunawan, A. A. S. (2021). Implementation of wemos d1 for wi-fi based controller tank-based military robot. *ICIC Express Lett. Part B Appl, 12(4),* 377-382. http://doi.org/10.24507/icicelb.12.04.377.

[15] Omar, M. F., Termizi, A. A. A., Zainal, D., Wahap, N. A., Ismail, N. M., & Ahmad, N. (2016). Implementation of spatial smart waste management system in Malaysia. *IOP Conference Series Earth and Environmental Science, 37(1)*, 12059. http://doi.org/10.1088/1755-1315/37/1/012059.

[16] Rajput, S. K. and Patil, M. (2018). Waste Management in IoT- Facilitated Smart Cities- A Suvery. *International Journal of Computer Applications, 182(24),* 21. http://doi.org/10.5120/ijca2018918037.

[17] Hong, I., Park, S., Lee, B., Lee, J.-K., Jeong, D., and Park, S. (2014). IoT-Based Smart Garbage System for Efficient Food Waste Management. *The Scientific World Journal, 2014, 1.* https://doi.org/10.1155/2014/646953.

[18] Juwariyah, T., Krisnawati, L., & Sulasminingsih, S. (2021). Design of IoT-Based Smart Bins Integrated Monitoring System Using Blynk. *IOP Conference Series Materials Science and Engineering, 1125(1),* 012078. http://doi.org/10.1088/1757-899X/1125/1/012078.

[19] Alourani, A., Ashraf, M. U., & Aloraini, M. (2025). Smart waste management and classification system using advanced IoT and AI technologies. *PeerJ Computer Science, 11:* e2777. https://doi.org/10.7717/peerj-cs.2777.

[20] Alabdali, A. M. (2025). Blockchain based solid waste classification with AI powered tracking and IoT integration. *Scientific Reports, 15(1),* 15197. https://doi.org/10.1038/s41598-025-97030-2.

[21] Martinez, A. and Fenandez, E. (2013) Learning ROS for Robotics Programming. Birmingham, UK: Packt Publishing.

[22] Fernandez, E., Crespo, L. S., Mahtani, A., and Martinez, A. (2015) Learning ROS for Robotics Programming - Second Edition. Birmingham, UK: Packt Publishing.

[23] Li, S. Z., Chan, K. L., and Wang, C. (2000) Performance evaluation of the nearest feature line method in image classification and retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11)*, 1335-1339. http://doi.org/10.1109/34.888719.

[24] Forthofer, R. N., Lee, E. S., and Hernandez, M. (2007) Logistic and Proportional Hazards Regression. *Biostatistics: A guide to design, analysis and discovery, 2nd ed., ed. RN Forthofer*, 387-419.

[25] Li, M., Wang, M., and Wang, C. (2010) Research on SVM classification performance in rolling bearing diagnosis. *International Conference on Intelligent Computation Technology and Automation, 3,* 132-135. https://doi.org/10.1109/ICICTA.2010.747.

[26] Behzad, M., Asghari, K., Eazi, M., and Palhang, M. (2009) Generalization performance of support vector machines and neural networks in runoff modeling. *Expert Systems with Applications, 36(4)*, 7624-7629. https://doi.org/10.1016/j.eswa.2008.09.053.

[27] Webb, G. I. (2011) Naive Bayes: Feature Selection Ensemble Methods. Bioinformatics, pp. 13-31.

[28] Massaro, A., Maritati, V., and Galiano, A. (2018) Data Mining Model Performance of Sales Predictive Algorithms Based on Rapidminer Workflows. *International Journal of Computer Science and Information Technology, 10*, 39-56.

[29] Raschka, S. (2018) Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning, [Online]. Available: http://arxiv.org/abs/1811.12808. https://doi.org/10.48550/arXiv.1811.12808.

[30] Xu, Y., and Goodacre, R. (2018) On Splitting Training and Validation Set: A Comparative Study of CrossValidation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning. *Journal of Analysis and Testing, 2(3),* pp. 249-262. https://doi.org/10.1007/s41664-018-0068-2.

[31] Sutikno, T., Handayani, L., Stiawan, D., Riyadi, M.A., and Subroto, I. M. I. (2016) WhatsApp, viber and telegram: Which is the best for instant messaging. *International Journal of Electrical and Computer Engineering, 6(3)*, 909-914. https://doi.org/10.11591/ijece.v6i3.10271.

[32] Refaeilzadeh, P., Tang, L., Liu, H., Angeles, L., and Scientist, C. D. (2020) Encyclopedia of Database Systems. Encyclopedia Database Systems.

[33] Asmono, R. T., Wahono, R. S., and Syukur, A. (2015) Absolute Correlation Weighted Na¨ıve Bayes for Software Defect Prediction. Journal of Software Engineering, Vol. 1, No. 1, pp. 38-45.

[34] Hashim, R. T. (2018) A Comparative Study of Indexing using Oracle and MS-SQL Server for Relational Database Management Systems. *International Journal of Computer Science and Mobile Computing, 7(12),* 341-350.

[35] Gyor¨odi, C., Gyor¨odi, R., Andrada, I., and Bandici, L. (2016) A Comparative Study Between the Capabilities of MySQl Vs. MongoDB as a Back end for an Online Platform. *International Journal of Advanced Computer Science and Applications, 7(11),* 73-78.