Rotten Fruit Detection using IOT

Nur Safi Syuhrah Binti Abdul Rani
University Kuala Lumpur
Malaysian Institute of Information Technology
Menara Bandar Wawasan, PO 50250 Kuala Lumpur
safi.rani@s.unikl.edu.my

***Abstract*—This paper presents an IoT-based system designed to detect and predict fruit spoilage in real time, helping to reduce household food waste. The system uses an MQ4 gas sensor to monitor methane levels and a DHT22 sensor for temperature and humidity. These sensors are connected to a Wemos D1 ESP8266 microcontroller, which transmits data via the MQTT protocol to a Node-RED dashboard. A prediction model built in RapidMiner classifies the fruit status as Fresh, Overripe, or Rotten using gas readings and days of storage. Real-time notifications are sent to users via Telegram. The model achieved a prediction accuracy of 91.09%. The system is cost-effective, scalable, and suitable for household use, with future potential for integration into smart storage solutions**

Keywords— ***IoT, Fruit Spoilage Detection, MQTT, Node-RED, Machine Learning, RapidMiner, Telegram Bot***

I. INTRODUCTION

In the era of smart homes and Internet of Things (IoT) advancements, maintaining food freshness has become a critical challenge, especially in households where fruits are easily forgotten and left to spoil. Traditional methods for fruit monitoring are reactive, relying on visual inspection or expiration dates, which often result in unnoticed spoilage and increased food waste. The growing emphasis on sustainable living and smart home integration necessitates a more proactive and intelligent solution.

This project aims to revolutionize how households monitor and manage fruit storage by developing an IoT-based spoilage detection system that provides real-time status updates. By using sensor technology and machine learning, the system identifies early signs of fruit decay and notifies users before the fruit becomes inedible.

The primary objectives of this project are:

- To use MQ4 gas sensors to get gas reading for rotten fruits by setting baseline readings in a fresh, non-spoiling environment.
- To develop an IoT-based system using sensors to continuously monitor the condition of fruits and detect early signs of spoilage, such as changes in gas emissions or other environmental factors.
- To create an alert system that notifies users when fruits are at risk of rotting, based on the data gathered from IoT sensors.

The significance of this system lies in its ability to minimize household food waste, improve produce management, and support sustainable living through smart technology.

Figure I.1 shows the system prototype of the IoT-based Rotten Fruit Detection System, illustrating the integration of sensors, data processing, and real-time notification components

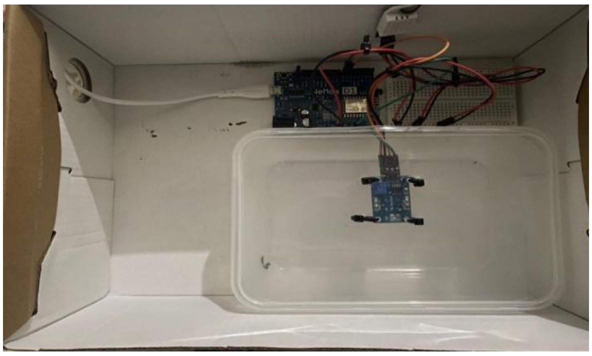

Figure I.1 The Prototype

II. LITERATURE REVIEW

The integration of IoT and machine learning technologies for food spoilage monitoring has gained significant attention in recent years. Karthickeyan et al. [1] introduced a freshness detection system combining CNN, YOLO, and gas sensors (MQ2 and MQ135), demonstrating that multimodal data inputs significantly enhance the accuracy of food quality assessment. Their architecture, although effective, requires image data and extensive computational resources, making it less feasible for low-cost household deployment.

Saha et al. [2] developed an IoT-based fruit lifespan detection system using MQ4 and DHT11 sensors, coupled with machine learning for spoilage stage prediction. Their work focuses on warehouse environments and emphasizes the role of continuous environmental monitoring. This directly supports our approach of using similar sensors to predict spoilage but refines the application for home use.

Manjulamna et al. [3] explored the combination of gas sensing and UV sterilization for food preservation. While effective in reducing microbial spoilage, their system involves energy-intensive UV activation, limiting its practicality in everyday consumer environments. Our project avoids such complexity by relying solely on real-time sensing and predictive alerts.

In another study, Gaikwad et al. [4] proposed a cost-effective IoT model using MQ sensors and Blynk for fruit freshness tracking. Though efficient in alerting users when thresholds are exceeded, their system lacks the ability to classify ripeness stages. Our use of RapidMiner enables data-driven prediction rather than static thresholding.

Persis and colleagues [5] introduced a machine learning-enhanced IoT platform to classify banana ripeness using SGP30 and SHT40 sensors. They leveraged advanced algorithms such as CatBoost, highlighting the feasibility of using ML models for freshness evaluation. Inspired by this, we apply a Decision Tree model for its simplicity and interpretability within Node-RED.

These studies highlight the relevance and growing demand for smart spoilage detection systems. However, most prior work targets industrial or commercial contexts. This project fills the gap by tailoring a smart, low-cost, and real-time monitoring system for household use, offering practical utility and contributing to food waste reduction efforts.

III. METHODOLOGY

A. System Architecture

The Rotten Fruit Detection using IoT consists of five main layers :

1. *Sensor Layer*: Collects environmental and gas emission data using MQ4 and DHT22 sensors.
2. *Edge Device Layer*: Wemos D1 ESP8266 reads sensor data and sends it via Wi-Fi.
3. *Communication Layer:* Utilizes MQTT protocol (Mosquitto broker) to transmit data to Node-RED.
4. *Local Processing Layer*: Node-RED executes a Python script with a RapidMiner-trained model to analyze and predict fruit ripeness.
5. *Presentation Layer*: Displays sensor readings and prediction results on the Node-RED dashboard and sends alerts via Telegram.

Figure II.1 illustrates the block diagram of the system architecture, showing the interaction between these layers.

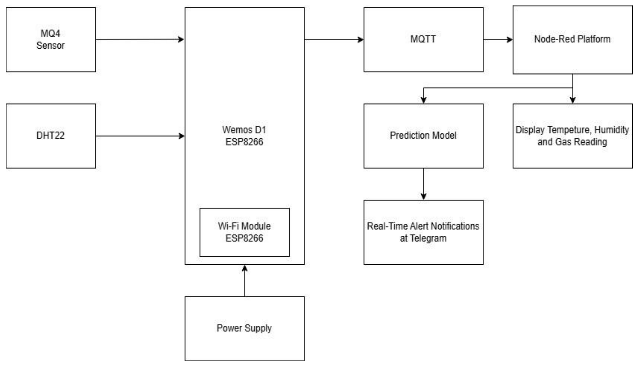

Figure III.1 Block Diagram

B. Hardware Components

The system utilizes the following hardware:

1. *Wemos D1 ESP8266*: Microcontroller with Wi-Fi capability.
2. *MQ4 Gas Sensor*: Detects methane gas emitted by decaying fruits.
3. *DHT22 Sensor*: Measures ambient temperature and humidity.

C. Software Components

Software tools used in this project include:

1. *Arduino IDE*: For programming and uploading code to the ESP8266.
2. *Node-RED*: For real-time data flow processing and dashboard visualization.
3. *RapidMiner*: To train and export the Decision Tree prediction model.
4. *Python 3*: Executes the trained model in Node-RED using the .pkl file.
5. *MQTT (Mosquitto Broker):* For efficient, lightweight communication.
6. *Telegram Bot API*: For sending real-time alerts and handling user commands.

D. Data Acquisition and Processing

To develop the fruit ripeness prediction model, data acquisition was carried out using seven types of fruits: banana, mango, tomato, cucumber, strawberry, blueberry, and grapes. Each fruit was stored in a typical household environment to simulate real-life storage conditions.

Gas emission levels were measured using the MQ4 sensor, while ambient temperature and humidity were recorded using the DHT22 sensor. Data collection was performed at regular intervals, with each fruit monitored daily over a period ranging from 7 to 20 days, depending on the fruit's natural ripening cycle. The recorded attributes included:

- Fruit Type
- Day
- Gas Reading
- Ripeness Status

All collected data was organized into a structured CSV file to support model training and evaluation in subsequent stages.

E. Machine Learning Development

The system employs a single classification model to determine the ripeness status of fruits based on sensor data:

1. *Classification Model*: Utilizes a Decision Tree Classifier to categorize the fruit condition as *Fresh*, *Overripe*, or *Rotten* based on gas emission values and the number of days since storage.

The Decision Tree algorithm was chosen for its interpretability, simplicity, and effectiveness in handling small-to-medium-sized datasets. It is particularly suitable for scenarios where model transparency is important and real-time decision-making is required, such as in IoT-based applications.

The training process for the classification model involves the following steps:

1. *Data Preprocessing*: Organizing raw data collected from the MQ4 sensor into structured CSV format. Each record includes the day number, gas reading, and corresponding ripeness label based on visual inspection.

2. *Feature Engineering*: Selecting “Day” and “Gas Reading” as input features, which together capture the spoilage progression over time.

3. *Model Training*: Feeding the labeled dataset into RapidMiner and applying the Decision Tree algorithm to learn patterns that distinguish between the three ripeness categories.

4. *Validation*: Using Cross Validation in RapidMiner to evaluate the model’s ability to generalize to unseen data, helping reduce overfitting and assess prediction reliability.

5. *Model Export*: Saving the trained model as a .pkl file using the Execute Python operator, enabling integration with external systems.

6. *Deployment*: Incorporating the trained model into the Node-RED flow using a Python script. The script is executed through the Exec node whenever new sensor data is received, allowing for real-time fruit ripeness classification.

IV. RESULT AND DISCUSSION

A. Data Acquisition and Processing

During the system testing phase, data was successfully collected from seven different fruit types: banana, mango, tomato, cucumber, strawberry, blueberry, and grapes. Each fruit was observed in a real household environment, and sensor readings were taken daily throughout its ripening cycle, covering durations between 7 and 20 days depending on the fruit type.

The MQ4 gas sensor captured the changes in gas concentration as the fruits progressed from fresh to rotten, while the DHT22 sensor continuously recorded the surrounding temperature and humidity. All readings were manually labeled with a ripeness status based on visual inspection and physical condition at each stage.

A total of 103 data entries were compiled into a structured CSV file, with four main attributes: Fruit Type, Day, Gas Reading, and Ripeness Status. The variation in gas emission across different fruit types and spoilage stages was clearly reflected in the recorded data.

The dataset was reviewed for completeness and consistency before being used for model training. This data served as a reliable foundation for developing the machine learning model and was later applied for real-time prediction within the Node-RED system

B. Machine Learning Model Performance

The prediction model was developed using RapidMiner with a Decision Tree algorithm to classify fruit ripeness into three categories: Fresh, Overripe, and Rotten. The dataset used for training consisted of 219 labeled entries across different fruit types and spoilage stages.

After preprocessing and assigning the appropriate roles to each attribute the model was evaluated using a 10-fold cross-validation approach within RapidMiner to assess its accuracy and generalization performance.

The model achieved an overall accuracy of 91.08% ± 8.75%, demonstrating strong performance in classifying fruit conditions. Precision and recall scores were highest for the Fresh and Overripe classes, while classification for the Rotten category showed lower accuracy, likely due to class imbalance in the dataset.

The trained model was exported as a .pkl file and integrated into the Node-RED environment using a Python script. During system testing, the model responded effectively to real-time sensor inputs and consistently produced accurate classification results, triggering appropriate Telegram notifications when spoilage was detected.

accuracy: 91.09% +/- 8.75% (micro average: 91.18%)

	true Fresh	true Overripe	true Root	class precision
pred. Fresh	62	1	1	96.88%
pred. Overripe	4	26	1	83.87%
pred. Root	1	1	5	71.43%
class recall	92.54%	92.86%	71.43%	

Figure IV.1 Performance Vector

C. Real-Time Monitoring and Visualization

The Node-RED dashboard provides a real-time view of the fruit storage environment, displaying essential metrics such as gas concentration, temperature, and humidity. This allows continuous monitoring of the fruit condition during storage and supports timely observation of any environmental changes.

In addition to the dashboard, users can interact with the system through a Telegram bot. By sending the */status* command, users can receive the latest sensor readings directly to their mobile devices, enabling convenient remote monitoring without accessing the dashboard interface.

Figures below show the Node-RED dashboard and the Telegram bot response to a /status command.

Figure IV.2 Node-RED Dashboard

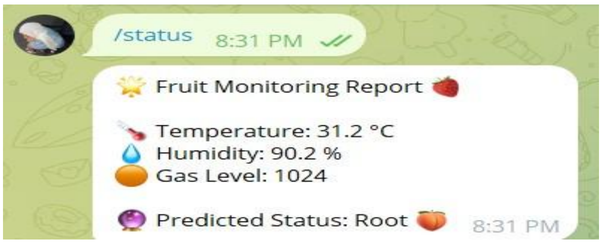

Figure IV.3 Telegram Command

D. Alert System Performance

The alert system was integrated using Telegram to notify users when the fruit condition reached either the Overripe or Rotten stage. This functionality was implemented in Node-RED through a combination of condition-checking function nodes and a Python-based prediction script. When the model output indicated a spoilage condition, a predefined message containing the fruit status, gas reading, and timestamp was sent directly to the user via Telegram.

During system testing, the alert feature functioned reliably and in real time. Messages were triggered immediately upon detection of Overripe or Rotten classifications, allowing users to take timely action. The response time between data input and alert delivery was typically under two seconds, ensuring minimal delay between spoilage detection and notification.

Figures below show examples of Telegram alerts automatically triggered by the system when a fruit was classified as Overripe or Rotten.

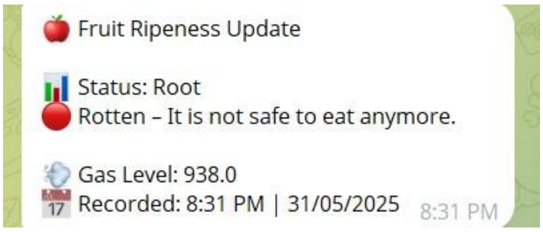

Figure IV.4 Rotten Alert

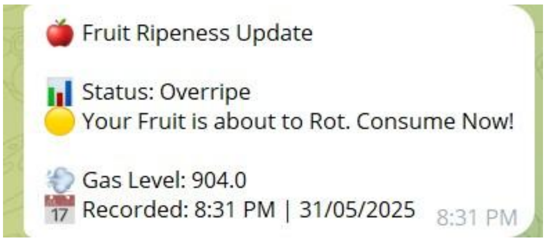

Figure IV.5 Ovverripe Alert

V. Conclusion

The Rotten Fruit Detection using IoT demonstrates reliable performance in real-time monitoring and classification of fruit conditions. By integrating environmental and gas sensing, lightweight communication protocols, and machine learning-based prediction, the system offers an effective solution for reducing fruit spoilage in household and potential commercial storage environments.

Key achievements of the system include:

1. Accurate classification of fruit ripeness using a Decision Tree model with an overall accuracy of 91.08%

2. Real-time sensor data acquisition and visualization using MQTT and Node-RED

3. Immediate Telegram alerts for Overripe and Rotten fruit conditions

4. User-friendly interaction through both dashboard monitoring and Telegram command access

5. Cost-effective implementation using commonly available IoT component.

These outcomes highlight the effectiveness of combining IoT and machine learning to enhance food monitoring and reduce waste. The system successfully provides timely and actionable information that allows users to intervene before spoilage occurs, improving food quality and storage efficiency.

Future work may include:

1. Expanding the model to include additional fruit types and larger datasets for improved generalization

2. Enhancing prediction accuracy by incorporating temperature and humidity as additional input features

3. Developing a multi-class confidence scoring system to better handle edge cases between ripeness stages

4. Integrating a local storage feature or cloud backup to track historical fruit conditions

5. Exploring advanced classification techniques such as Random Forest or Neural Networks for deeper insights

In conclusion, this project demonstrates the practicality and value of applying IoT and machine learning for smart fruit storage monitoring. As food sustainability becomes increasingly critical, systems like the one presented here can play an important role in minimizing waste, improving storage decisions, and contributing to smarter living environments..

ACKNOWLEDGEMENT

In the name of Allah, the Most Beneficent, the Most Merciful. It is with deep gratitude and appreciation that I express my sincere thanks to my supervisor, Assoc. Prof. Ts. Dr. Haidawati binti Mohamad Nasir, for her unwavering guidance, support, and encouragement throughout the course of my Final Year Project. Despite her demanding schedule, she consistently provided valuable insights and constructive feedback, which played a crucial role in the successful completion of this project. I would also like to extend my appreciation to Ts. Faridah binti Yahya and Sir Azmi bin Ahmad for their efforts in coordinating the Final Year Project and providing the necessary guidelines and assistance throughout the process. My heartfelt thanks also go to my beloved parents and family. Their endless support, patience, and encouragement have been the backbone of my journey. Whenever I felt overwhelmed or discouraged, their presence and motivation gave me the strength to carry on. This project would not have been possible without the help and support of all these individuals, and I am deeply grateful to each of them.

REFERENCES

[1] KarthickeyanP, B., Devi, K., & Professor, A. (2022). IOT Based Food Freshness Detection Using Deep Learning Techniques.

[2] Saha, A., Ali, L., Rahman, R., Hossain, T., Mohammad Bin Shamim, S. J., Supto, M. A. A., Shidujaman, M., Uddin, M. R., & Hasan, M. (2023). IoT Based Fruit Quality Inspection and Lifespan Detection System. 2023 7th International Conference On Computing, Communication, Control And Automation, ICCUBEA 2023.https://doi.org/10.1109/ICCUBEA58933.2023.10392254

[3] Manjulamna, P., Sai Sangeetha, G., Thaseen, S., Meghana, B., Madhuri, K. N. V., & Babitha, K. (2024). IOT-BASED FOOD SPOILAGE DETECTION SYSTEM WITH UV STERILIZATION. In International Journal of Research in Engineering, IT and Social Sciences (Vol. 14). http://indusedu.org

[4] Gaikwad, D. A., Pophale, A. D., Palande, S. D., Baviskar, K. S., & Pirjade, N. J. (n.d.). FRUIT FRESHNESS DETECTION AND MONITORING USING IOT. www.irjmets.com

[5] Persis, V. (n.d.). *Developing an IoT and ML-driven Platform for Fruit Ripeness Evaluation and Spoilage Detection: A Case Study on Bananas*. https://ssrn.com/abstract=4838544