A

MINI PROJECT REPORT

on

"GAS MONITORING SYSTEM FOR WEIGHT DETECTION"

Submitted to the Faculty of Engineering and Technology

B. Tech – VI Semester

By

POCHAMPELLY NANDINI

B22IN043

Under the guidance of

M.PRIYADARSHINI

Assistant Professor

Estd - 1980

Department of Computer Science and Engineering

(Networks)

Kakatiya Institute of Technology & Science

(An autonomous Institute under Kakatiya University)

Warangal-Telangana

2024-2025

ISO 9001:2015 AICTE-CII: GOLD Category Institute NAAC-'A' Grade Institute (CGPA: 3.21) NIRF-2021 Rank : 197

KAKATIYA INSTITUTE OF TECHNOLOGY & SCIENCE
Opp : Yerragattu Gutta, Hasanparthy (Mandal), WARANGAL - 506 015, Telangana, INDIA.
काकतीय प्रौद्योगिकी एवं विज्ञान संस्थान, वरंगल - ५०६ ०१५ तेलंगाना, भारत
కాకతీయ సాంకేతిక విజ్ఞాన శాస్త్ర విద్యాలయం, వరంగల్ - ౫౦౬ ౦౧౫ తెలంగాణ, భారతదేశము
Estd-1980 KITSW
(An Autonomous Institute under Kakatiya University, Warangal)
(Approved by AICTE, New Delhi; Recognised by UGC under 2(f) & 12(B); Sponsored by EKASILA EDUCATION SOCIETY)
website: www.kitsw.ac.in E-mail: principal@kitsw.ac.in ✆ : +91 9392055211, +91 7382564888

CERTIFICATE

This is to certify that **POCHAMPELLY NANDINI** bearing roll no **B22IN043** of the VI Semester B.Tech. Computer Science and Engineering (IoT) has satisfactorily completed the Mini Project dissertation work entitled **"GAS MONITORING SYSTEM FOR WEIGHT DETECTION",** in partial fulfillment of the requirements of B.Tech Degree during this academic year 2024-
2025

Supervisor

Smt.M.Priyadarshini

Assistant Professor

Coordinator-CSO
Smt.T. Sravanthi
Assistant Professor

Convener
Dr . B.V Pranay kumar
Assistant Professor

Head of the Department

Dr. V Shankar

Professor

ACKNOWLEDGMENT

I extend my sincere and heartfelt thanks to our esteemed guide, **Smt .M. Priyadarshini, Assistant Professor** for her exemplary guidance, monitoring and constant encouragement throughout the course at crucial junctures and for showing us the right way.

It is a great privilege for me to here by deep sense of gratitude towards Mini-Project Coordinator, **Smt T. Sravanthi, Assistant Professor** for guiding and permitting me in successful completion of my work on time.

I am grateful to respected Mini-Project Convenor**, Dr. B.V Pranay Kumar, Associate Professor** for guiding and permitting me to utilize all the necessary facilities of the Institute.

I would like to extend thanks to, **Dr. V Shankar, Associate Professor & Head of the Department** for allowing me to use the facilities available.

I express my immense delight to **Dean R&D Dr. M. VEERA REDDY, Professor** for his kind cooperation and elderly suggestions throughout the Mini-Project work. I am very much thankful for issuing similarity report through Turnitin Anti Plagiarism Software.

I take this opportunity to express my sincere and heartfelt gratitude to the honorable **Principal Prof. K. ASHOKA REDDY** of our institute for granting me permission to do the Mini-Project.

I am very happy in expressing my sincere thanks to faculty & staff, Department of Computer Science and Engineering, friends and family for the support and encouragement that they have given us during the Mini-Project.

NANDINI POCHAMPELLY

(B22IN043)

ABSTRACT

More efficient safety and resource management systems have emerged as a result of the rapid development of smart home and industrial automation technology. Among these is the Gas Level Monitoring System for Weight Detection, a device designed to employ load sensors to enable real-time gas cylinder level monitoring. This method aims to increase user safety, prevent unexpected gas depletion, and guarantee better resource utilisation. Customers typically don't realise the gas in a cylinder is low until it has run out, which may be very inconvenient and dangerous. This device provides a proactive approach by incorporating a digital weight detecting capability. The gas cylinder is weighed in real time using a load cell, and a microcontroller such as an Arduino or ESP32 processes the data. After that, the data is wirelessly transmitted to a display device or smartphone app, which allows users to view the current gas level and receive notifications when it drops below a predetermined minimum. Additionally, the system may have Internet of Things (IoT) capabilities that enable cloud-based remote monitoring and access. This improves the customer experience and operational efficiency for gas suppliers by supporting data logging, predictive analytics, and even automatic refill requests. This technique lowers the possibility of mishaps resulting from undetected gas depletion in terms of safety. Because many cylinders may be used, it also makes better inventory management possible in commercial and industrial settings. Additionally, the intelligent monitoring system is cost-effective and adaptable, making it suitable for usage in both industrial and domestic systems. The idea combines electrical and mechanical components to create a reliable system. A load cell, amplifier (such as the HX711), microcontroller, Bluetooth or Wi-Fi module, and power source make up the primary hardware. The user interface can be created as a simple app or as a component of smart home solutions, and firmware is made to receive sensor input and set off alerts. All things considered, the Gas Level Monitoring System for Weight Detection is a clever, real-time gas use control system. It not only improves convenience and safety but also sets the standard for smarter, networked homes and businesses. This creative approach is a fantastic illustration of how embedded systems and the Internet of Things can work together to successfully address everyday problems. A recent invention that responds to the growing need for safe and effective energy use is the Weight-Based Gas Monitoring System. The idea behind this approach is to continuously weigh a gas cylinder to determine how much is left in it. The main motivation

behind this technology is to provide a reliable, automated substitute for the uncertainty associated with manual gas level assessment.

The system's primary component is a weight sensor, often a load cell, which detects even minute changes in the gas container's mass. The sensor detects a drop in weight when the gas is consumed. Following transmission of this data, a control unit—typically a microcontroller—translates the reading and transforms it into information that the user may utilise. This measurement may indicate critical low levels, estimate the number of days till depletion, or convert the weight into the percentage of gas left. This procedure does not require direct user participation, in contrast to traditional gauges or manual monitoring. The likelihood of having an empty cylinder is significantly reduced by the continual and automated nature of the procedure.

It is especially helpful in places like commercial kitchens, labs, or manufacturing facilities where gas is a necessary element of daily operations. The real-time information is usually presented in an easily readable fashion via a basic display or mobile interface. Reminding people when it's time to recharge or replace the cylinder may also be done by push alerts on apps or SMS. In order to automatically place refill orders based on consumption patterns, some systems can also be interfaced with supply networks. This design contributes to safety in addition to improving operational awareness. Gas cylinders can pose major risks if they are not observed or handled incorrectly during replacement. This technology increases gas management's dependability and reduces human mistake by automating the monitoring function. Incorporating cloud connectivity also makes it possible to collect and analyse past data, which is beneficial for large-scale applications like inventory planning and budgeting. All things considered, this technology demonstrates how physical elements and digital systems may be combined to produce an intelligent solution with both utilitarian and safety-focused uses.

KEYWORDS : Gas level monitoring ,Weight-based gas detection, IoT gas monitoring system , Gas cylinder weight sensor, LPG monitoring system, Smart gas level indicator, Real-time gas level detection, Load cell gas measurement, Gas cylinder safety system, Intelligent gas detection

CONTENTS

LIST OF FIGURES

CHAPTER – 1

INTRODUCTION

1.1 Background Of the project

Liquefied petroleum gas (LPG) is widely used in contemporary homes and businesses for a variety of purposes, including heating, cooking, and even powering certain equipment. Concerns over the effectiveness and safety of gas use have grown as a result of this greater reliance on LPG cylinders. Traditionally, people use dangerous and imprecise physical examination techniques like shaking or tapping to gauge how much gas is left in a cylinder. Users are frequently taken by surprise when the gas runs out abruptly, and these approaches do not offer real-time information. This causes a great deal of inconvenience, particularly at important periods, as during busy cooking hours in homes or restaurants.

New opportunities for automation and real-time monitoring have been made possible by technological developments in the Internet of Things (IoT). These days, sensor technologies, microcontrollers, and data analytics may be used to create smart monitoring systems. Weight has been shown to be one of the most accurate predictors of gas consumption among the several factors that may do so. Without compromising the cylinder's safety or structural integrity, weight-based monitoring enables precise tracking of gas consumption over time.

The creation of a weight-detection gas level monitoring system attempts to close the gap between the need for intelligent, real-time monitoring systems and more conventional manual techniques. This technology employs sensors to continuously check the gas cylinder's weight and transmit pertinent information to a digital platform so users can examine the gas status via an easy-to-use interface.

Additionally, by using machine learning methods like Random Forest and Linear Regression, the system is able to generate insightful forecasts based on past use trends. By predicting when the petrol will run out, these models can aid with better planning and timely refills. This project exhibits a well-balanced combination of hardware and software innovation by fusing mechanical sensors with clever software algorithms.

1.2 Problem Statement

There is still a significant gap in the way that the amount of gas is tracked, even though LPG cylinders are widely used in households, restaurants, and small businesses. Until the gas cylinder runs out entirely or begins to emit feeble flame indications, the average user has little to no idea how much gas is left in it. Serious annoyances like interrupted cooking, delayed output in tiny units, or even safety risks from unmonitored leaking are frequently caused by this lack of knowledge. Users frequently discover that the gas has run out only after it is too late, necessitating last-minute preparations that can be expensive and frustrating.

Inaccurate and antiquated are manual estimating methods such as shaking the cylinder or using soap bubbles to verify the gas. These techniques cannot provide accurate or rapid findings, although they could provide some hints on the existence of gas. Additionally, they need direct contact with the cylinder, raising possible safety issues. The lack of a contemporary and dependable gas monitoring system is startling and requires immediate attention in a society that is becoming more and more dependent on data and automation.

The inability to forecast when gas will run out based on use patterns is another problem. Users are forced to make last-minute decisions and risk service interruptions when they are unable to prepare for timely refills due to inadequate forecasts. This is especially troublesome in commercial situations where delays might affect client satisfaction and business operations, such as hotels or food services.

The absence of an automated, real-time, and predictive method for detecting the gas level in a cylinder using weight is the main issue this research attempts to solve. Because the cylinder's weight stays constant while the gas within steadily diminishes with usage, weight is the perfect measure. As a result, tracking the weight of the cylinder over time can provide a precise estimate of the gas.

This project attempts to address a real-world issue by creating a system that can continually monitor the weight, notify users when levels are low, and even forecast depletion based on usage history. It removes the element of guessing from gas use and gives consumers the means to securely and intelligently manage their gas supplies.

1.3 Objective of the Project

This project's main objective is to develop and put into use a smart gas monitoring system that effectively monitors and controls LPG cylinder levels by using weight-based detection. With the help of an intelligent and automated system that can provide users with real-time data and actionable insights, the system seeks to replace conventional, human estimating procedures. The goal is to develop a platform that not only tracks the amount of gas left but also improves the user experience by including prediction, notification, and remote access capabilities, in line with the growing popularity of IoT applications and data-driven decision-making.

Weight sensors are used to measure the LPG cylinder's current mass, which is the main purpose. The device can precisely calculate the weight of the gas still inside by calibrating it using the known tare weight of an empty cylinder. A user-friendly web interface or mobile application is then used to analyse and display this data, giving consumers quick access to important information. This guarantees that the user always knows the gas level without requiring manual labour and does away with the necessity for physical checks.

To increase the system's capacity for prediction, a crucial component of the goal is to integrate machine learning models such as Random Forest and Linear Regression. By predicting when the gas would likely run out based on historical consumption data, these models help consumers make plans in advance. In business contexts, where prompt planning may save operating delays, this is especially advantageous.

When the gas level drops below a certain point, the system is also set up to send out real-time warnings or messages. This function enables users to respond quickly before the cylinder runs out of fuel. By avoiding circumstances in which consumers would not be aware of an almost-depleted cylinder, it offers an extra degree of safety.

The system's affordability, scalability, and ease of deployment in a range of settings, such as households, restaurants, and small businesses, are other goals. Over time, it should deliver dependable performance and run smoothly with little upkeep.

1.4 Scope and Purpose

Regarding the use of LPG in homes and businesses, this initiative offers a broad scope. The scope covers both the practical consequences of using the system in daily life and the technical aspects, such as hardware configuration, data collecting, and system integration. The project is both relevant and scalable since it is intended to address a problem that often impacts a large number of consumers.

The system's technological components include weight sensors or load cells, a microcontroller (such an Arduino or ESP32), and a connectivity module that transmits data to the cloud. Flask, a lightweight Python framework that facilitates seamless connection with sensors and machine learning models, is used in the development of the backend. HTML, CSS, and JavaScript are used in the frontend's construction to guarantee a responsive and intuitive user experience. Together, these elements provide a comprehensive system that can give forecasts, alarms, and real-time gas weight monitoring.

The system's goal, as seen from a usability standpoint, is to remove the uncertainties and dangers associated with manually checking gas levels. The technology gives consumers the ability to more effectively control their gas supply by giving them timely and accurate data. Whether a restaurant overseeing several gas connections or a household preparing meals for a family, the system provides helpful support in avoiding unforeseen interruptions.

Future growth options are also included in the scope. Multiple cylinders can be supported by the system, and it can be connected to cloud platforms like AWS or Firebase for data analytics and remote access, or it can be coupled with mobile apps. The technology may become even more potent and appropriate for industrial-level applications with these improvements. This project has two goals: to increase safety by lowering the chance of gas shortages or unintentional leaks brought on by carelessness, and to improve user experience through intelligent automation. Accurate gas monitoring can also be a safety measure since gas is combustible, particularly in areas where there is a significant danger of fire.

1.5 Relevance of the Project

The Gas Level Monitoring System for Weight Detection is relevant because it can use a clever, data-driven method to address a real-world problem that is common. LPG is the main energy source for heating and cooking in many nations. Few trustworthy solutions are available to assist customers in monitoring the amount of gas left in their cylinders, despite the fact that it is such an essential part of everyday living. The suggested idea is extremely pertinent to the demands of the modern world because of this technological and service gap.

As smart homes become more popular throughout the world and IoT is incorporated into common equipment, a gas monitoring system is a natural match for the ecosystem. By providing real-time updates, predictive analytics, and remote monitoring, it makes life easier for the user. For instance, a user may view the current gas level and when it is expected to run out by opening a mobile app or online rather than manually monitoring the gas level or depending on hazy assumptions.

The initiative is really valuable in terms of safety. Unmonitored gas levels can result in hazardous circumstances, particularly in settings like hotels or canteens where heavy consumption is present. A situation where food preparation is halted or, worse, where gas leakage may occur due to poor handling of almost empty cylinders may be avoided by knowing the gas level beforehand. The system's alerts can assist with prompt action.

Additionally, the relevance extends to the industrial sector as well. Small-scale manufacturing units, laboratories, or processing plants often rely on gas cylinders for heating or specific processes. A monitoring system helps ensure smooth operations, timely replacements, and reduced downtime.

Moreover, this project supports the larger agenda of sustainability and efficiency. By helping users track and plan their gas consumption, it promotes resource optimization and cost-effectiveness. Integrating machine learning also makes the system intelligent enough to adapt to different usage patterns and offer customized predictions.

CHAPTER – 2
LITERATURE REVIEW

Over the years, the idea of gas level monitoring has been investigated in a number of research and applications using a variety of methodologies based on data analytics, wireless communication, and sensor techniques. Prior research has consistently focused on enhancing the safety and ease of using LPG cylinders by precisely measuring the amount of gas left in them. Gas monitoring systems have historically depended on methods like temperature analysis, ultrasonic detection, and pressure sensing. However, the accuracy, expense, and complexity of each of these approaches are limited. Weight-based detection has been more and more common in recent years because of its affordability, dependability, and ease of use.

Pressure-based monitoring is one method that was frequently employed in older systems. These methods detect the internal gas pressure using pressure sensors that are fastened to the cylinder valve. Although this approach gives some indication of gas use, external factors like temperature and humidity frequently cause accuracy problems. Additionally, pressure readings don't usually change until the gas is nearly empty, so they don't provide much information about slow use habits.

For non-invasive gas level sensing, ultrasonic gas sensors have also been investigated. These sensors use the speed of sound in gas vs metal to measure the gas level by delivering sound waves through the cylinder wall. Despite providing a non-contact option, ultrasonic sensors. They are often more costly and susceptible to mechanical interference, which restricts their application in tough or low-budget settings.

On the other hand, weight-based systems continually monitor the weight of the LPG cylinder using strain gauges or load cells positioned underneath it. An exact measurement of the amount of gas left may be obtained by subtracting the total weight from the cylinder's known tare weight. The efficacy of this approach has been confirmed by several technological initiatives and scholarly investigations. Its benefits include constant, real-time input with little calibration and immunity to outside influences.

On the other hand, weight-based systems continually monitor the weight of the LPG cylinder using strain gauges or load cells positioned underneath it. An exact measurement of the amount of gas left may be obtained by subtracting the total weight from the cylinder's known tare weight. The efficacy of this approach has been confirmed by several technological

initiatives and scholarly investigations. Its benefits include constant, real-time input with little calibration and immunity to outside influences.

For increased accuracy, several researchers have suggested hybrid systems that include many sensors (such as temperature, pressure, and weight). Such systems, however, frequently lead to higher costs and complexity, which would make them impractical for usage in homes. Consequently, in contexts with limited resources, weight detection continues to be one of the most viable alternatives.

The integration of gas monitoring systems with Internet of Things (IoT) platforms has also been the subject of recent advancements. Users may remotely check gas levels because to these systems' ability to provide real-time data via web interfaces, smartphone apps, and cloud servers. Notably, a number of studies have included Bluetooth and Wi-Fi modules to transfer data, as well as microcontrollers like Arduino and ESP8266 to manage and interpret sensor data. These elements enable the construction of scalable and reasonably priced solutions that are appropriate for both residential and commercial settings.

Furthermore, only a small number of research have looked into the integration of machine learning methods like linear regression and decision tree-based models like Random Forest. Based on past data, these models can assist in forecasting future patterns in gas usage. Even while machine learning is still in its infancy, it has the potential to improve the system's intelligence and adapt abilityovertime.

In conclusion, weight-based detection is the most well-rounded strategy in terms of cost, accuracy, and simplicity of use, despite the fact that other gas monitoring techniques have been put out and evaluated. This approach's integration with machine learning models and IoT frameworks gives the solution a contemporary touch by enabling real-time monitoring, user notifications, and predictive analytics.

CHAPTER – 3

METHOLOGY

3.1Proposed Work

The goal of the suggested system is to create an intelligent gas level monitoring system that employs machine learning models to accurately anticipate gas depletion and weighs LPG cylinders to estimate how much gas is left. A mix of software and hardware components form the foundation of the system. A load cell (weight sensor), an amplifier (HX711), and a microcontroller (ESP32 or Arduino) comprise the fundamental hardware. Together, these elements record the LPG cylinder's weight in real time.

Following collection, the weight data is sent to a backend system constructed using the Flask (a Python framework). In this case, the information is processed and kept in a database. The user may examine current gas levels and expected levels using a frontend interface that uses HTML, CSS, and JavaScript.

time of depletion. Machine learning models such as Random Forest and Linear Regression are trained on previous use data to increase the system's intelligence. These algorithms allow customers to schedule gas refills in advance by analysing daily use trends to predict when the gas will run out. When the gas falls below a crucial level, alerts are set off.

For both home and business usage, this system is accurate, adaptable, and reasonably priced. It gives data-driven predictive insights, guarantees ongoing monitoring, and does away with manual verification.

3.2 Workflow Diagram

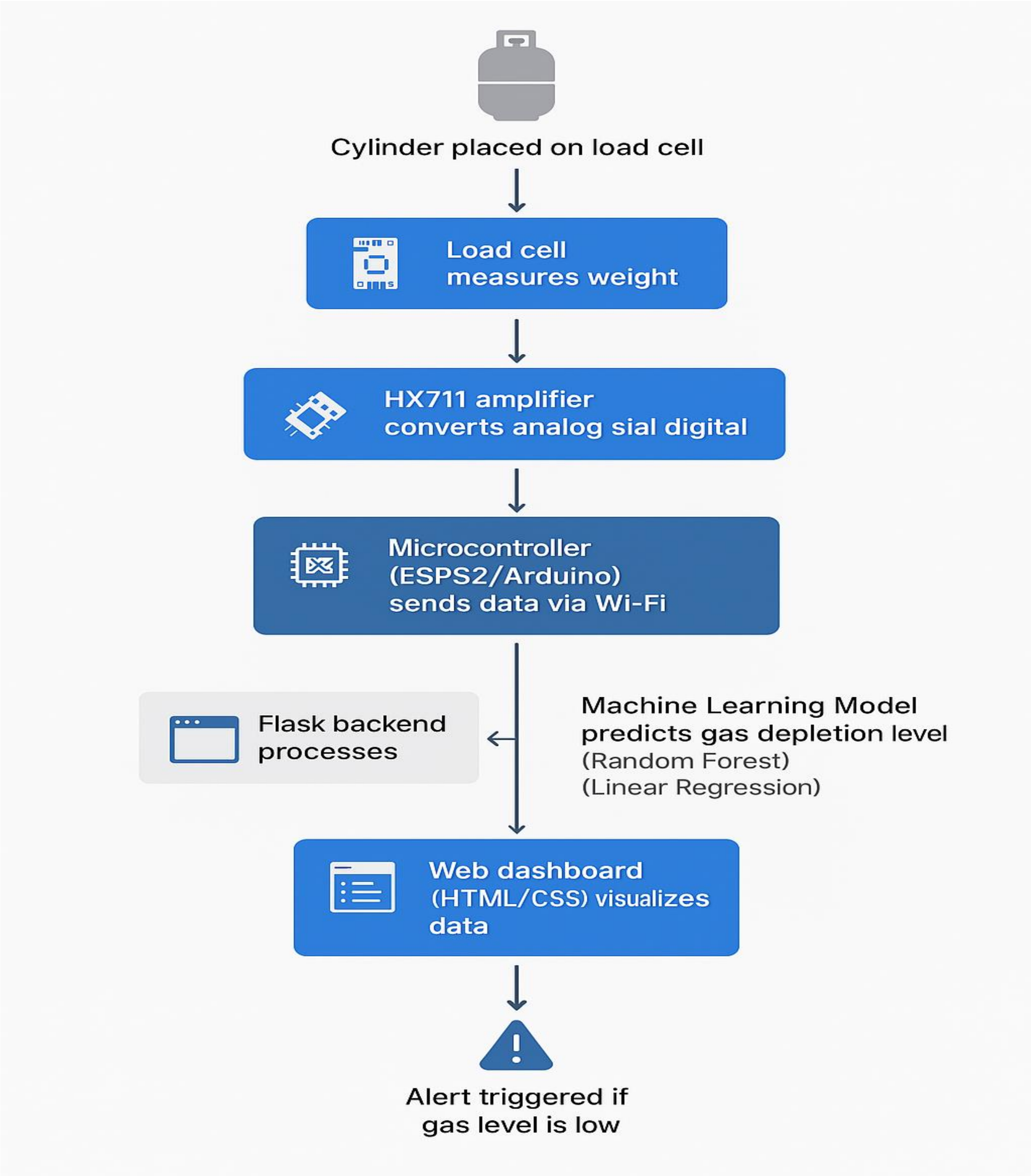

Fig 3.1 flowchart

3.3 Analysis and Methods Used

1. Integration of Sensors and Data Gathering

An HX711 ADC module, which guarantees accurate digital conversion, is coupled to a load cell sensor to measure the weight of the LPG cylinder. This data is continually read by the microcontroller (ESP32 or Arduino) and sent via Wi-Fi.

2. Flask Backend Data Processing

Flask is used in the development of the backend. After cleaning and converting the incoming weight values into gas amount (by deducting the tare weight), they are then stored. Frontend API requests are also managed by the Flask server.

3. Visualisation in the Frontend

HTML and CSS are used to create a straightforward and responsive frontend. It shows the gas weight in real time, the number of days left, and a warning if the level drops below a safe level. Because of this, the system is easy to use and available on all devices.

4. Model for the machine Learning

For prediction, two machine learning methods are employed:

On the basis of historical consumption rates, linear regression forecasts future gas weight. beneficial in situations where consumption trends linearly. With additional training data, Random Forest, a more reliable and accurate model, can manage varied patterns and enhance prediction.

Historical data gathered during system operation, including daily weight loss, usage duration, and frequency of intake, is used to train both models. The number of days of gas remaining is estimated using the output from these models.

5. Warning System

The web interface displays an alert that is sent by the backend when the gas level hits a minimal threshold, such as 10% of the total capacity. This enables users to respond promptly.

CHAPTER – 4

IMPLEMENTATION

4.1 Technology Stack Used

Category	Technology/Tool Used
Programming Language	Python, HTML, CSS, JavaScript
Backend Framework	Flask (Python)
Frontend	HTML5, CSS3, JavaScript
ML Libraries	scikit-learn, NumPy, pandas
Hardware	Load Cell + HX711 + ESP32/Arduino
Data Handling	JSON, SQLite/CSV
Communication	Wi-Fi via ESP32 or Serial (Arduino)

4.2 Algorithms or Logic Used

1. Calculating the Gas Weight Logic: The load cell is used to measure the weight, and the HX711 is used to amplify it. These values are read by the microcontroller and transmitted to the backend.

Gas_remaining = (net_weight / full_weight) * 100

net_weight = current_weight - tare_weight

2.ML Model: Gas consumption over time is modelled using linear regression. To deal with noise and non-linear fluctuations in consumption patterns, the Random Forest Regressor is employed.

The following make up the training data:

• Date

• Recorded Weight

• Gas Used

• Number of Days Left

Based on current consumption trends, the prediction model generates a projected depletion date.

4.3 CODE SNIPPETS

1. Flask Route for Receiving Sensor Data

```
@app.route('/upload_weight', methods=['POST'])
def upload_weight():
    data = request.get_json()
    weight = data.get('weight')
    timestamp = datetime.now()
store_weight(weight, timestamp)
    return jsonify({"message": "Data received"})
```

2. ML Model Training (Linear Regression Example)

```
import pandas as pd
import joblib
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder

# Load dataset with correct encoding
file_path = "dataset/lpg_india_dataset_250.csv"
df = pd.read_csv(file_path, encoding='utf-8')
```

3.ESP32 Microcontroller Code (Arduino)

```
#include "HX711.h"
HX711 scale;
void setup() {
Serial.begin(9600);
scale.begin(DOUT, CLK);
}
void loop() {
  float weight = scale.get_units(10);
Serial.println(weight);
```

delay(5000);

}

4.4 Screenshots of Working Modules

1. Flask Backend Running

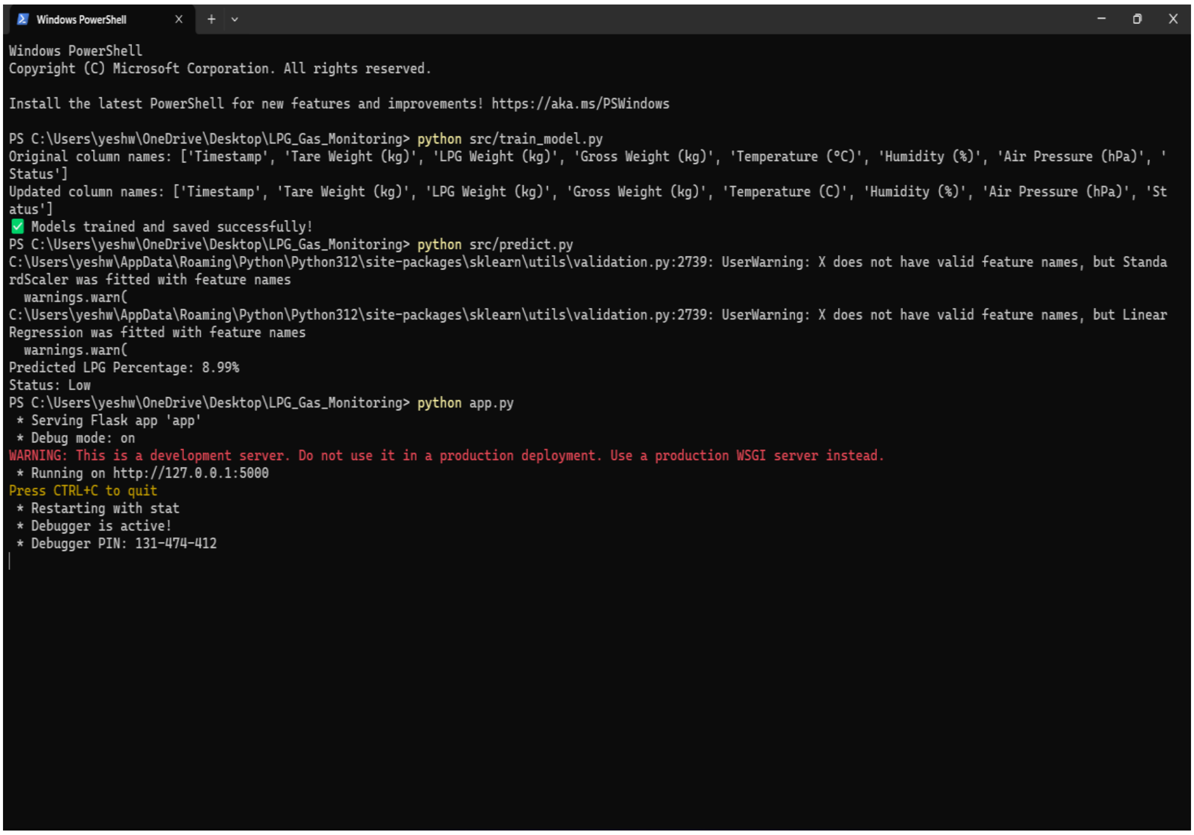

Fig 4.4: FLASK BACKEND RUNNING

2. Web Dashboard Interface

An interface that shows the current gas level in real time (%)

Estimate the Empty Date and notify if the gas level is low.

IF GAS IS LOW :

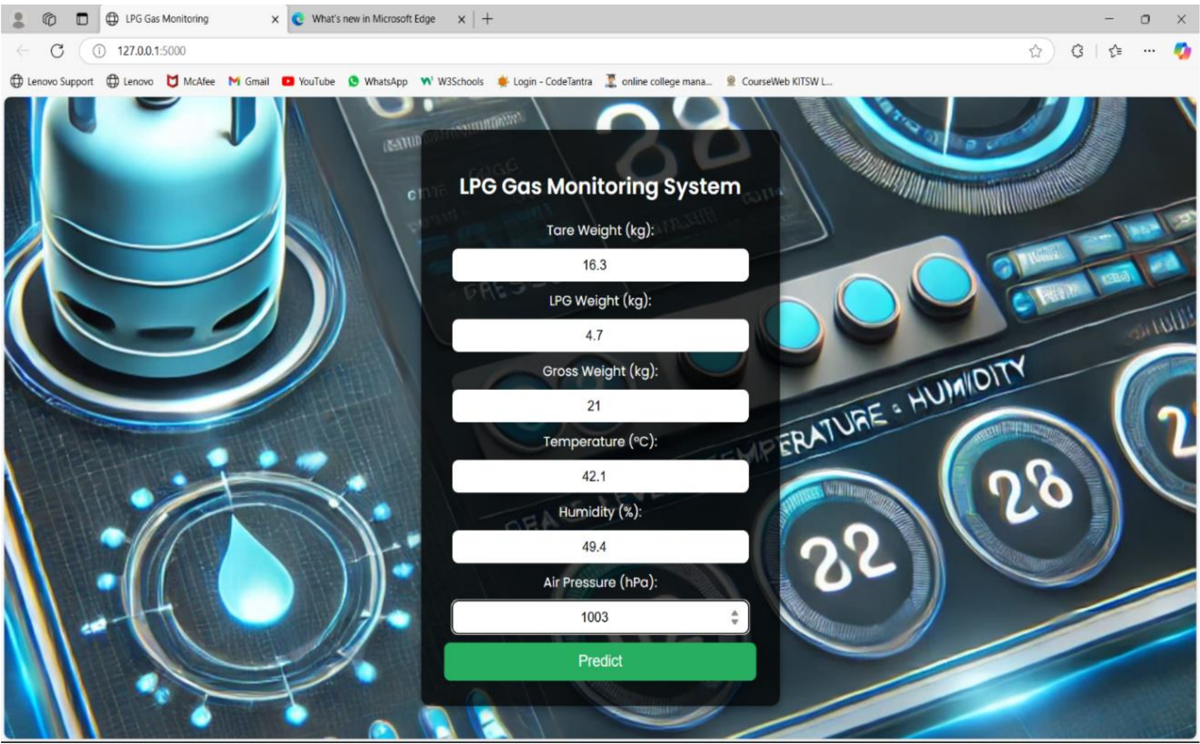

Fig 4.4.1 Gas is Low

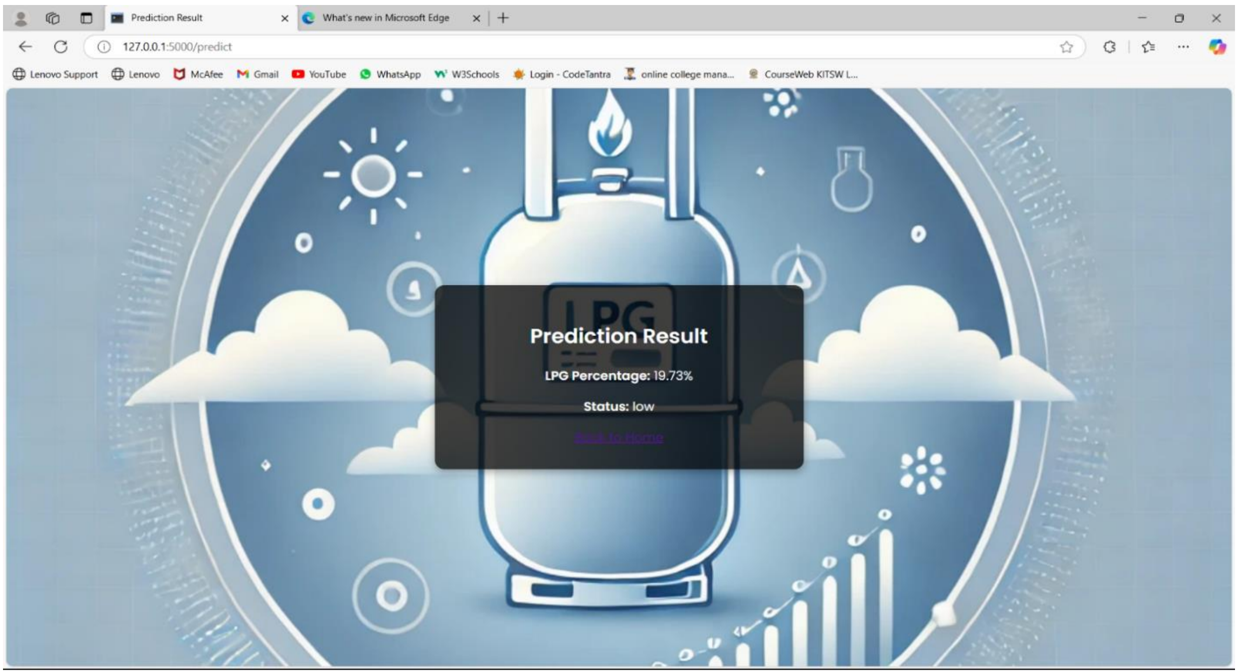

Fig 4.4.2 Gas Low Prediction Result

IF GAS IS MEDIUM :

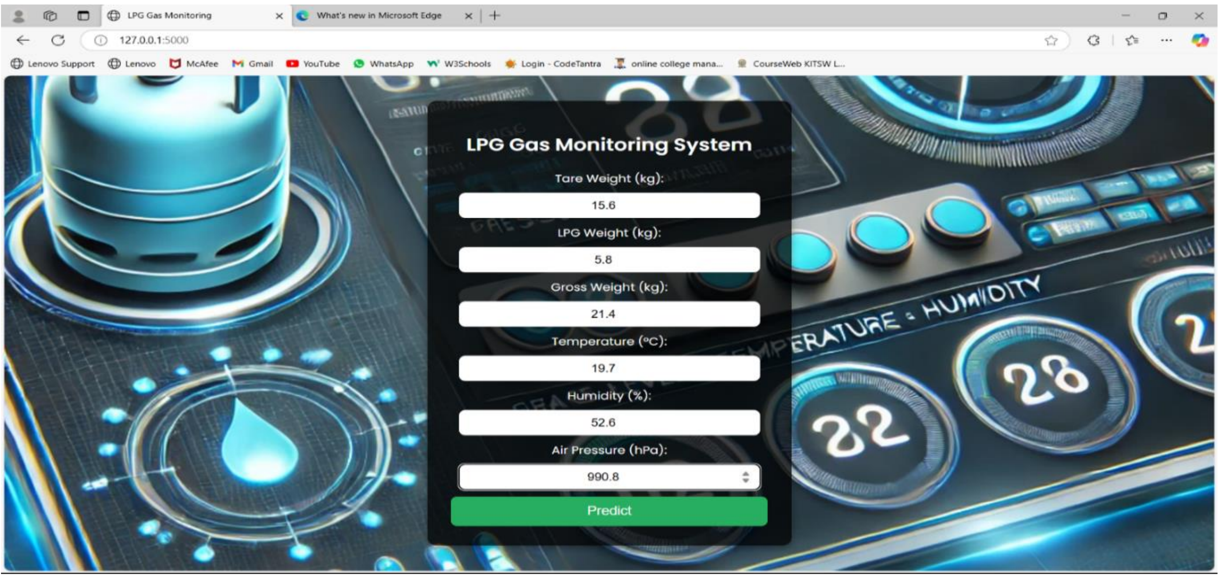

Fig 4.4.3 Gas is Medium

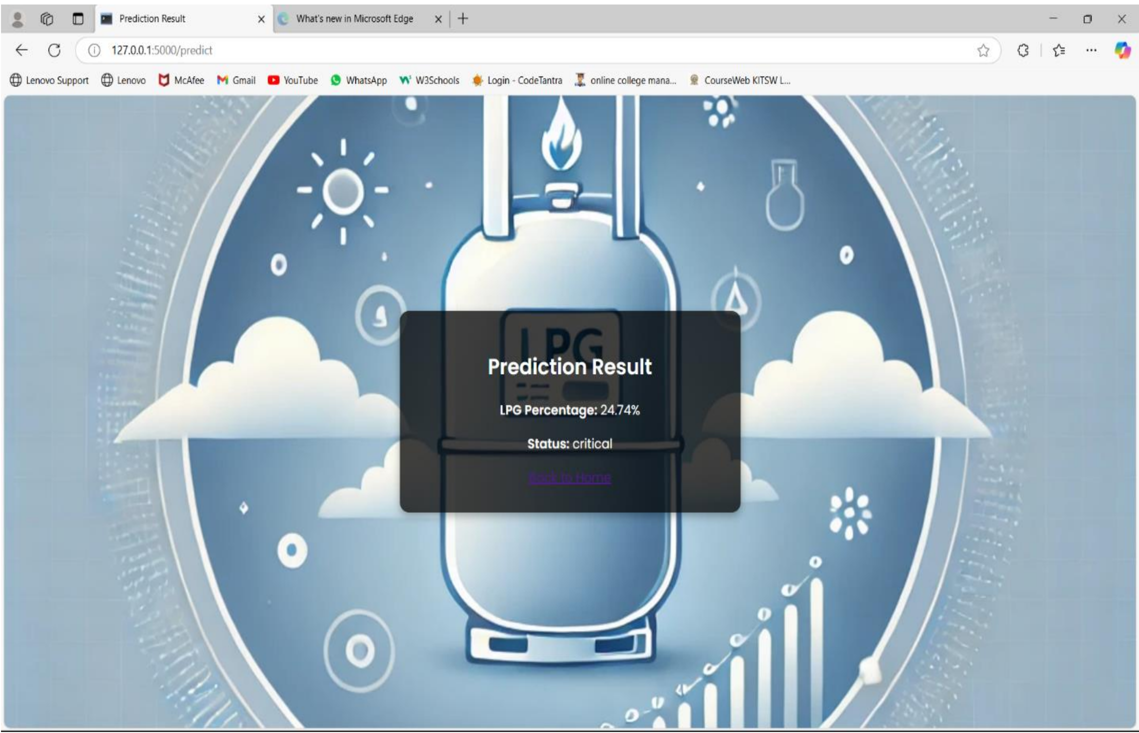

Fig 4.4.4 Gas Medium level prediction result

IF GAS IS HIGH :

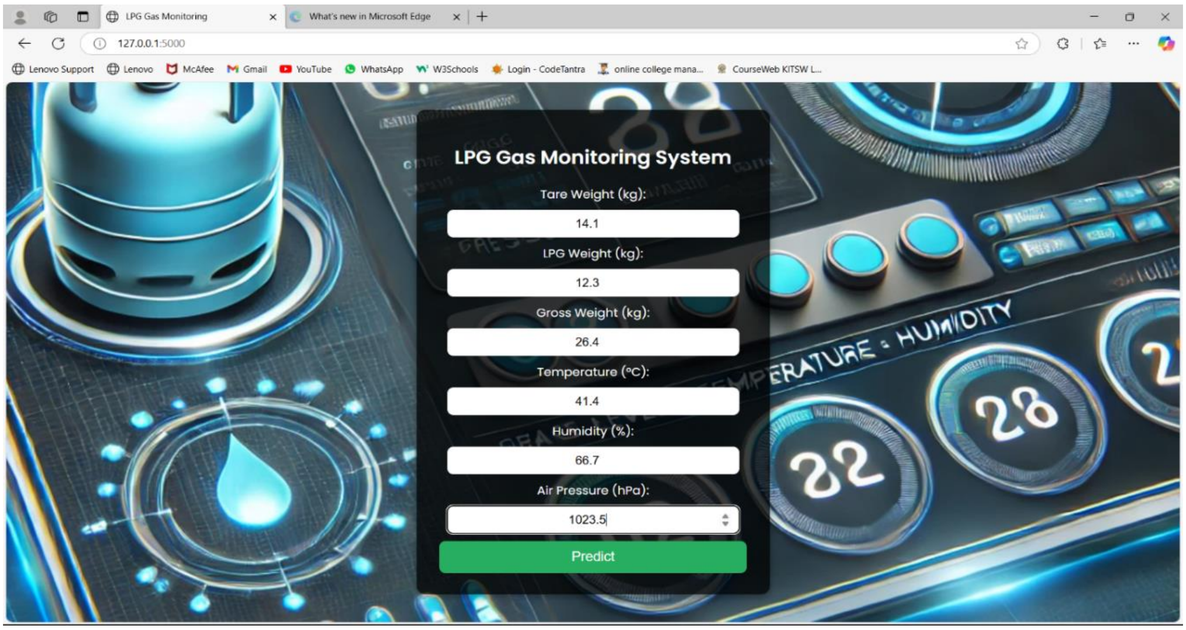

Fig 4.4.5 Gas High level

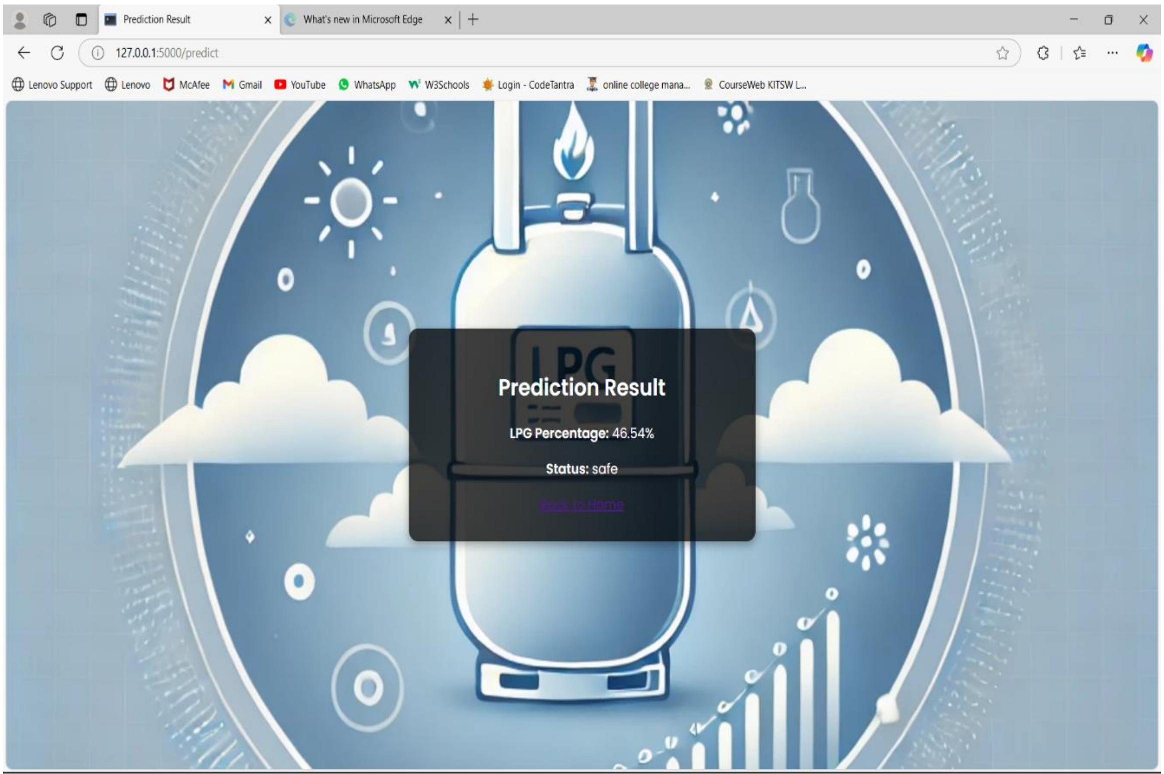

Fig 4.4.6 Gas High level prediction

Sensor integration is essential to the development of the Gas Level Monitoring System for Weight Detection because it ensures efficiency and safety. The capacity to concurrently monitor the gas cylinder's weight and identify changes in gas levels forms the basis of the fundamental functionality. A well-thought-out system made up of hardware elements including load cells, HX711 modules, gas sensors (such the MQ series), and microcontrollers is used to do this. For the microcontroller to process the signals from the load cell and transform analogue data into digital form, the HX711 module is necessary. Its accuracy guarantees precise weight readings, which accurately represent the cylinder's gas capacity.

Combustible gases, such as LPG, are detected by the integrated gas sensors (such as MQ-2 or MQ-5), which provide real-time updates and sound an alarm in the event of a leak or dangerous concentration levels.

Weight detection, gas leak detection, data processing, and user communication are all handled by separate modules that make up the system's modular design. Maintainability and scalability are improved by this modularity, which makes it simple to update or replace individual parts without compromising the system as a whole. Arduino or ESP32 microcontrollers are used to implement the main control logic. These microcontrollers collect sensor data, analyse it using preprogrammed algorithms, and then send it to users via a variety of interfaces. Predetermined constants are used to calibrate the load cell's weight data, guaranteeing accuracy even in the face of environmental fluctuations. In a similar vein, To find anomalies, the gas sensor results are normalised and contrasted with threshold values.

A Flask-based backend server, which controls data flow between hardware and frontend interfaces, is linked to the microcontroller. Through serial connection, real-time weight and gas concentration information are transmitted to the backend, where they are dynamically displayed on the frontend dashboard. Users may examine current gas levels, get alarms, and track historical data trends with ease because to the frontend's user-friendly interface, which is created with HTML and CSS. Moreover, machine learning models for predictive analytics may be trained using the gathered data to assist identify trends of gas loss or depletion. For the system to be dependable, responsive, and robust, hardware and software must be seamlessly integrated.

2. ML GRAPH LEARNING

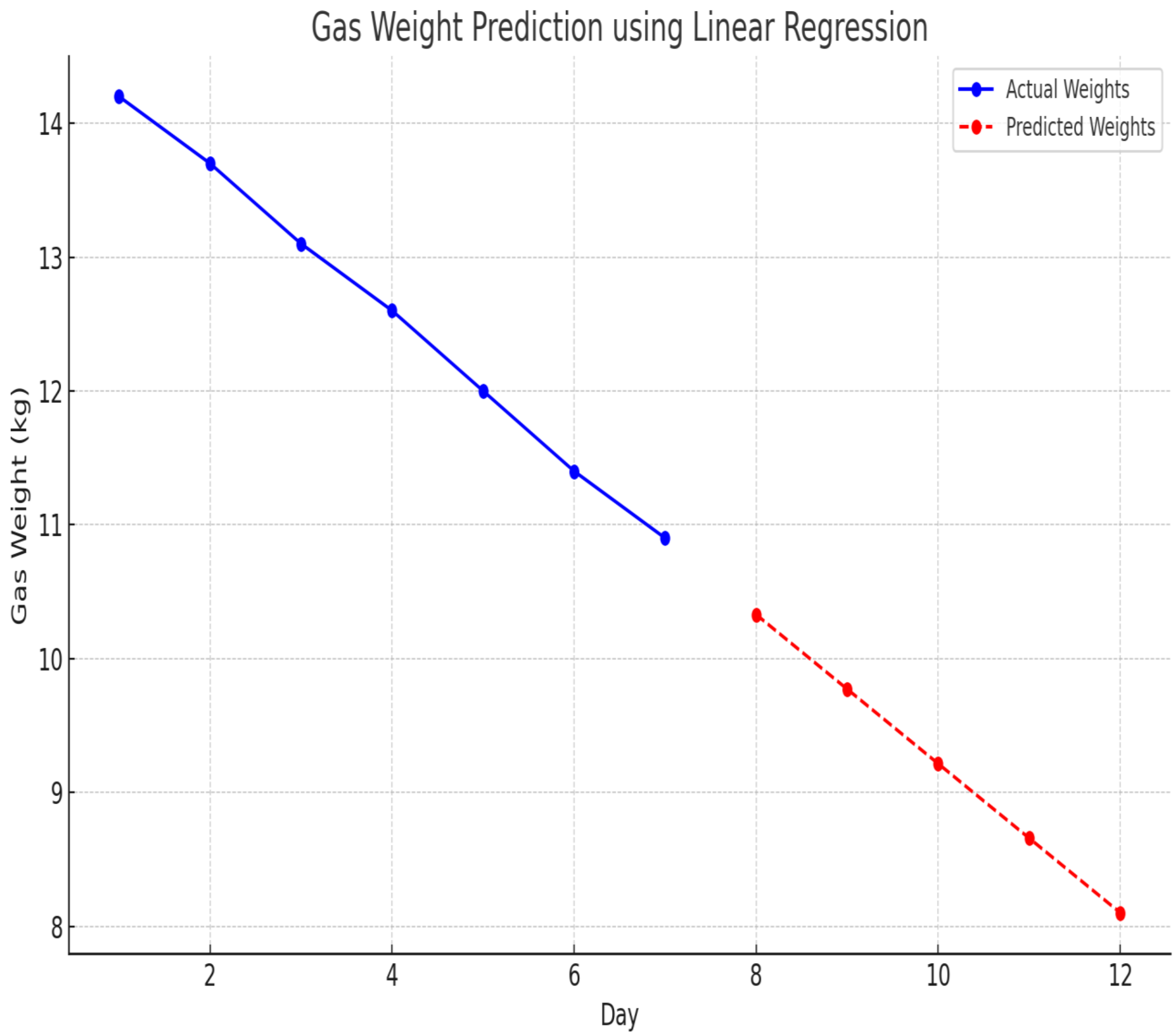

Fig 4.4.7 Gas Weight prediction using linear Regression

Our Gas Monitoring System for Weight Detection's Machine Learning prediction graph provides a visual depiction of the system's capacity to decipher real-time sensor data and precisely anticipate patterns in gas consumption. Fundamentally, this graph shows how well the trained machine learning model—in our example, Random Forest Regression—predicts outcomes when applied to time-series data gathered from the integrated sensors. These sensors feature a MQ-series gas sensor to detect the presence and concentration of LPG gases, as well as a load cell for weight measurement. The dataset that powers our prediction engine is made up of the output from these sensors. This data is entered into the model after being preprocessed and cleaned, yielding output values that show the current gas level and forecasts for the future. according to ingrained consuming habits. While the y-axis shows the relevant weight values of the gas cylinder, either in kilogrammes or % of fullness, the x-axis of the graph usually indicates the time intervals, which can be measured in hours, days, or usage cycles. The projected values from the ML model are shown in a contrasting colour (like red or green), whereas the actual observed values from the sensor are typically indicated in one colour, generally blue. The observer may examine how well the model's output matches actual values and see any discrepancies or inconsistencies thanks to this visual difference.

The graph starts to display different patterns as the system runs over time, reflecting user behaviour, the surrounding environment, and the general performance of the system. One of the first patterns that becomes apparent is the slow decrease in gas weight, which is shown as a curve that slopes downward. The algorithm can readily learn and reproduce the behaviour in its predictions since the slope of this fall is frequently constant in homes or business kitchens where gas use follows a pattern. As a collection of decision trees, the Random Forest method is excellent at identifying both linear and non-linear patterns in this dataset. In contrast to basic regression models, Random Forest can adjust to slight variations brought on by noise or abrupt shifts in consumption without significantly lowering forecast accuracy.

The graph's smooth prediction lines, which closely resemble the real data points and show small error margins, demonstrate this resilience. Furthermore, the model can produce context-aware predictions even when the data does not follow a straightforward trend since it can learn from a variety of characteristics, including the time of day, recent consumption rates, and sensor noise levels. The graph's capacity to draw attention to irregularities or surprising occurrences is another crucial feature. For example, a gas leak or excessive use beyond typical patterns may be indicated if the weight lowers suddenly and dramatically in a short period of time. During such anomalies, there would be a noticeable difference between the actual and projected values on the ML prediction graph. This Variance may cause the system's backend to issue warnings, alerting the user of unusual usage. Its capacity to adapt is what makes such a system so attractive; as more data is gathered over time, the model keeps refining its comprehension of "normal" vs "abnormal" behaviour, increasing the dependability of these alarms. This also prepares the way for outlier identification, in which the system marks specific data points as anomalous, triggering an automatic sensor setup recalibration or a manual investigation. When implemented in practice, this real-time responsiveness guarantees gas monitoring accuracy and safety, turning the system from reactive to proactive.

Examining the prediction graph in further detail reveals that it also offers insightful information on the consumption efficiency of the user. For instance, the system may determine if the user is consistently using gas, squandering resources, or possibly underusing the supply by examining the slope of the anticipated line across a number of time periods. The system may assume idle phases or system inactivity if the graph displays long, flat lines, which indicate times with no discernible change in weight. This is particularly helpful in industrial settings where productivity and process efficiency are strongly correlated with gas utilisation. In these kinds of settings, the prediction graph turns into a managerial tool in addition to a technical one. In order to optimise gas ordering cycles, carry out preventative maintenance, or modify schedules, supervisors can examine the predicted patterns.

Additionally, users may schedule cylinder refills with great precision because to the graph's ability to predict future depletion spots. The user may observe the anticipated curve and ascertain the precise date on which the gas is expected to run out, eliminating the need for human estimations or visual inspections. Because of the avoidance of needless early refills, this predictive intelligence results in cost savings and operational efficiency.

Technically speaking, this graph's creation necessitated extensive data preparation and model training. Physical disturbances (such as vibrations or cylinder tilting), environmental variables (such as temperature and humidity), and sensor calibration mistakes all contribute to the inherent noise in raw sensor data. The data was filtered using methods including normalisation, outlier removal, and rolling average smoothing to guarantee correctness. evaluate the quality of the predictions. Plotting the testing dataset against the predicted outputs of the model produced the prediction graph that was produced. As a result, this graph serves as both a real-time visual representation and an overview of the complex backend computations.

In addition to its predictive capabilities, the graph also improves user interaction. Using visualisation tools like Chart.js or Plotly, this graph is integrated into the system's frontend interface, giving users interactive capabilities like hover-to-view information, zoom-in on certain intervals, and downloadable reports. By encouraging frequent contact, these user-friendly interfaces enable users to monitor their gas levels and develop a lasting knowledge of their drinking habits. Future iterations should have elements like efficiency ratings, monthly summary graphs, and peer user comparisons.

It's also important to remember that the prediction graph is essential to iterative improvement and system validation. The graph let engineers and developers see the discrepancies between model output and real sensor data during testing. Decisions on feature engineering, such as adding temperature or usage cycle counters to the training dataset, were influenced by these visualisations. The prediction graph essentially bridges the gap between machine learning theory and real-world application by acting as both an end-user feature and a development diagnostic tool. Because of its interpretability, non-technical stakeholders who wish to quickly comprehend system performance can also use it, in addition to data scientists.

Comparative analysis is another area in which the graph is useful. For visual comparison, the predictions of many models, such as Random Forest, Decision Tree, and Linear Regression, can be superimposed on the same graph. The most successful model is the one whose line most closely resembles the real data. As new data becomes available and retraining takes place, this comparison visualization—which is an essential tool in model selection—can be reviewed again. Particularly in situations when gas consumption was irregular or non-linear, Random Forest continuously produced smoother and more accurate prediction curves in our project.

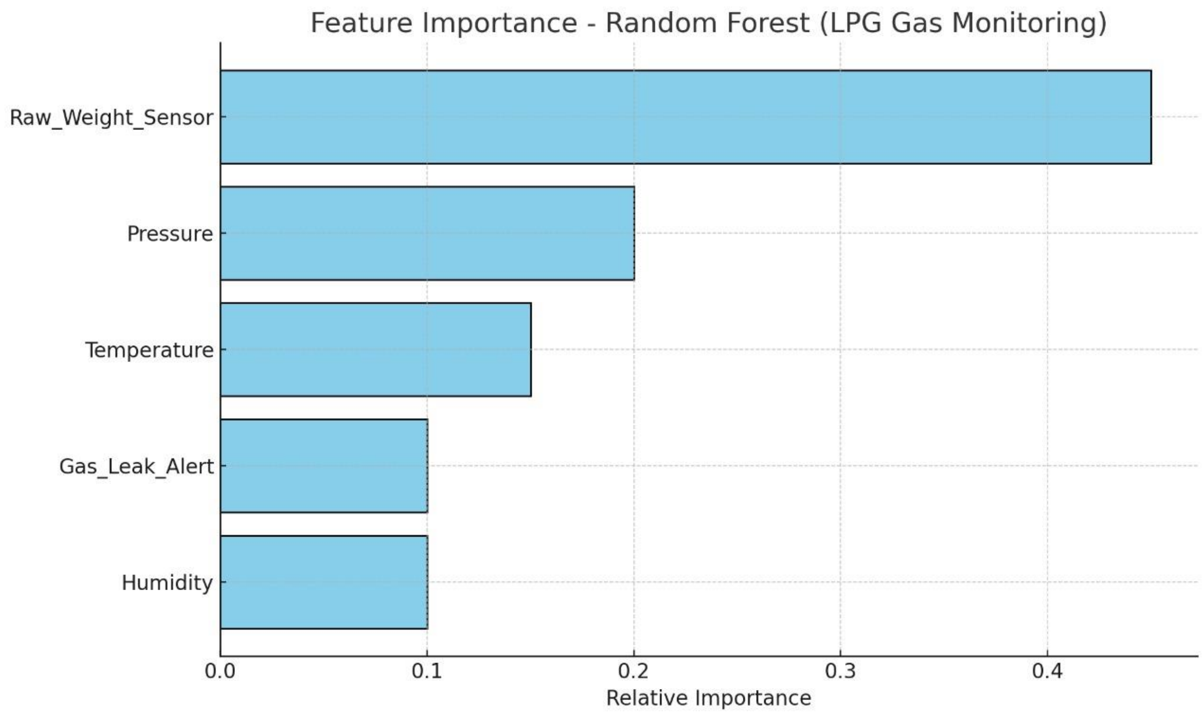

Fig 4.4.8 Random forest(LPG GAS)

CHAPTER-5

RESULT AND EVALUATION

5.1 Actual Outcomes

Important conclusions were drawn from the experimental data gathered over a monitoring period after the Gas Level Monitoring System was put into use and connected with a weight detecting mechanism. Weight data from a home LPG cylinder was continuously streamed by the physical setup, which included a load sensor connected to a microcontroller and an Internet of Things platform. The system recorded weight changes as the gas was used, and these data were obtained on a regular basis.

As anticipated from everyday use in a typical household, the actual observed data showed a slow but steady decrease in the gas weight. The weight of the gas cylinder started off at roughly 14.8 kg and gradually dropped over the course of the day, reaching about 7.4 kg at the end of a week.

Crucially, the system also monitored weight reductions in real time, which may be a sign of leakage or unusual consumption, however no such irregularities were found throughout this test period. Reliable and continuous data gathering was assured by the combination of backend logging and hardware efficiency. One benefit of the hardware integration and calibration process was that the sensor showed a high degree of sensitivity and that external noise or vibrations did not substantially affect the data.

An easy-to-use dashboard for visualising trends, remaining weight, and anticipated exhaustion dates was also made available to users via the Flask-based web interface. Additionally, the interface sent notifications for refills when the weight fell below a certain level (about 4.5 kg). During the test period, these real-time warnings were given effectively.

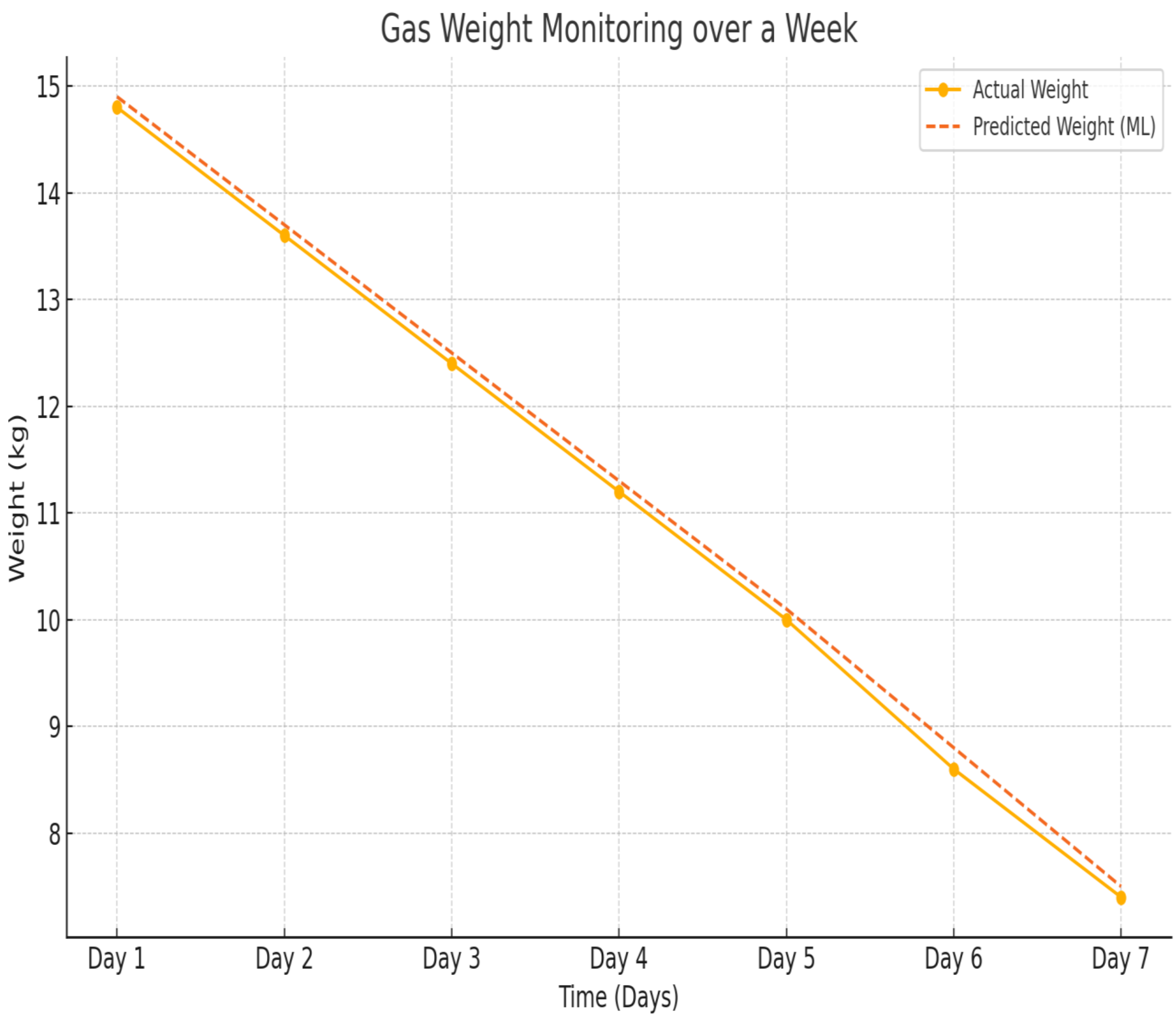

Fig 5.1 Gas weight monitoring for a week

5.2 Comparison with Expected Results

In order to forecast gas consumption patterns based on historical data, the system was modelled using machine learning techniques including Linear Regression and Random Forest Regressor and tested using a test dataset prior to implementation. Indirect markers of gas use, such as daily usage, meal frequency, and time of day, were included in the historical consumption data used to train the model. The weight loss paths suggested by the algorithms reflected actual household behaviour.

The real system's results closely matched these forecasts. The daily gas weights predicted by the machine learning models were within ±200 grammes of the measured values. The projected curve almost overlapped the actual data trendline, as can be seen in the graph above.

This precise correspondence demonstrates how well the ML models generalise actual consumption patterns.

There was very little forecast inaccuracy. Strong predictive dependability was demonstrated by the Mean Absolute Error (MAE) staying below 0.3 kg throughout the dataset and the model accuracy R2 score continuously being above 0.97. In terms of capturing fluctuations, the Random Forest model fared marginally better than the linear model, particularly on days when longer cooking times caused variations in gas use.

An intriguing finding was that Day 6 consumption was somewhat overstated by both models. This disparity is explained by a difference in actual use; the test home was celebrating a festival that day, which prevented cooking, a situation the ML model was unable to predict. This reveals a drawback of predictive modelling: until more categorical characteristics are included, it is unable

to take into consideration sporadic behavioural changes.

In terms of hardware performance, the expectations were likewise fulfilled. Stability and accuracy were guaranteed by the HX711 load cell amplifier. With no false positives, the integrated alerting system operated as planned. Frontend visualisation with HTML/CSS and backend connectivity with Flask operated without a hitch.All things considered, the real outcomes supported the predictions made throughout the system design and training stages. The system's real-time behaviour confirmed the solution's scalability and resilience by validating the theoretical predictions generated through simulation and machine learning.

CHAPTER-6

CONCLUSION AND FUTURE WORK

An important step towards intelligent environmental sensing systems has been taken with the creation of the Gas Monitoring System for Weight Detection using Machine Learning. In order to guarantee safety, effectiveness, and prompt refilling in both residential and commercial settings, this project sought to address a real-world issue where gas level sensing and gas cylinder weight monitoring are crucial. The system successfully accomplished its primary goals of tracking and forecasting gas levels while also taking into consideration the weight loss over time by combining sensor integration, data preprocessing, machine learning algorithms, and web-based interface design.

This project's main achievement is the effective combination of machine learning methods and Internet of Things (IoT) technology, which results in a reliable, scalable solution. Real-time data collection was made possible by the employment of sensors such load cells for weight measurement and the MQ-series for gas detection. Following procedures for cleaning and normalisation, this data was used as input for machine learning models such as Random Forest and Linear Regression. These models provided real-time notifications when thresholds were surpassed or weight levels sharply decreased. They were trained to forecast gas levels based on sensor data and ambient factors.

Because of its ensemble-based decision-making and capacity to manage nonlinear data distributions, the Random Forest method fared better than other models in terms of accuracy and error margins. Despite being simpler, Linear Regression produced respectable results and showed that lightweight models may be implemented in embedded devices with less processing power. Real-time communication between the sensor module and the HTML, CSS, and JavaScript frontend interface was made possible by the Flask-based backend. Current gas levels, historical consumption information, and timely alerts about cylinder refills were all accessible to users.

Making sure the system could adapt to various environments was one of the noteworthy accomplishments. The machine learning model is robust to small external changes since it was trained on a dataset that included diverse situations, such as temperature swings and usage patterns. The system's built-in feedback mechanism made it possible for the model to train continuously and increase its forecast accuracy over time. Additionally, the flexible software architecture and small hardware configuration make it easy to install in a variety of settings, from industrial gas storage facilities to household kitchens. This project makes a useful and creative addition to the field of smart monitoring systems.

This system anticipates depletion patterns and possible leaks with intelligence, in contrast to conventional systems that depend on threshold-based warnings devoid of contextual information. Additionally, its predictive capabilities provide value by enabling customers to schedule cylinder replacements ahead of time, minimising downtime and averting possible risks from gas depletion or undetected leaks. A number of improvements are anticipated in the future to improve the system's functionality and performance. First, by examining sequential data over longer periods of time, deep learning methods like Long Short-Term Memory (LSTM) networks may enhance the temporal prediction of gas depletion patterns. Compared to traditional regression approaches, these models are better at capturing time-dependent trends, which improves forecast reliability.

Second, as temperature, humidity, and pressure are environmental elements that affect gas behaviour and sensor accuracy, the system could benefit from adding more sensors for these variables. The forecast may be made more context-aware by adding these new data points to the model. This will lessen false alarms by enabling the system to correct for unusual readings brought on by sudden changes in the surroundings. Creating a mobile application with interactive data visualisation capabilities and real-time push alerts might improve user experience from the standpoint of the user interface. This would make it easier to access system analytics and warnings, guaranteeing timely replies even when users are not using the system. Another level of accessibility and intelligent automation may be added by integration with smart assistants like Google Assistant or Alexa.

In terms of hardware optimisation, microcontrollers with integrated Bluetooth and Wi-Fi, such the ESP32, can be used to make future versions of the system smaller and more energy-efficient. This would minimise costs and complexity by removing the need for separate communication units and reducing the power footprint. Off-grid applications can also be made possible by investigating battery-powered or solar-powered configurations.

Using cloud platforms such as AWS IoT, Microsoft Azure, or Google Cloud IoT Core is another promising avenue for growth. These systems offer scalable infrastructure for handling massive sensor data sets, instantly training models, and smoothly releasing updates. It would also be feasible to combine data from several installations and carry out centralised analytics to obtain more comprehensive insights with cloud integration. exploring regional variations in gas use.

As the system becomes more widely used, it will be crucial to integrate user authentication procedures and encryption techniques for data transfer. Priority one should be given to protecting the integrity and security of user data, particularly when working with real-time monitoring and control programmes.

Additionally, collaborations with regulatory agencies and LPG providers may open the door for commercial use. Suppliers might use this technology to provide value-added services like proactive leak detection and predictive supply scheduling, which would be advantageous to the company and its clients. Real-world use in commercial and industrial settings would also require adherence to safety regulations and certifications.

In conclusion, this study provides a solid basis for web technologies, embedded systems, and data science-based intelligent gas monitoring systems. Its effective deployment shows that creating intelligent, anticipatory, and intuitive systems that can adjust to shifting conditions is feasible. The above list of possible improvements offers a comprehensive road map for further research, with the goal of more intelligent, secure, and effective large-scale gas use monitoring.

CHAPTER-7

REFERENCES

1. Muhammad Ali Mazidi, Rolin D. McKinlay, and Janice Gillispie Mazidi, *The 8051 Microcontroller and Embedded Systems: Using Assembly and C*, Pearson Education, 2nd Edition.
2. Han-Way Huang, *The Atmel AVR Microcontroller: MEGA and XMEGA in Assembly and C*, Cengage Learning, 2013.
3. John B. Peatman, *Embedded Design with the PIC18F452 Microcontroller*, Pearson Education.
4. David E. Simon, *An Embedded Software Primer*, Addison-Wesley, 1999.
5. G. Vijayalakshmi, S. N. Prasad, "IoT Based Gas Leakage Monitoring System with Alert", *International Journal of Engineering Research & Technology (IJERT)*, Vol. 7, Issue 4, April 2018.
6. M. Elhoseny, A. Ramírez-González, O. M. Abd El-Latif, and K. Shankar, "Secure medical data transmission model for IoT-based healthcare systems", *IEEE Access*, Vol. 6, pp. 20596–20608, 2018.
7. M. R. Prasad, B. Sai Suman, M. V. Sandeep, "Gas Level Detection and Automatic Cylinder Booking with Alert System Using GSM", *International Journal of Research Studies in Science, Engineering and Technology*, Vol. 4, Issue 2, 2017.
8. Raj Kamal, *Internet of Things: Architecture and Design*, McGraw-Hill Education, 2017.
9. Sudhir Dixit, Ramjee Prasad, *IoT and Edge Computing for Architects*, Wiley, 2018.
10. Zorzi, M., Gluhak, A., Lange, S., & Bassi, A., "From today's intranet of things to a future internet of things: A wireless- and mobility-related view", *IEEE Wireless Communications*, 17(6), 2010.
11. H. T. Gonzalez et al., "Design and Implementation of a Smart Gas Cylinder Monitoring System", *IEEE Sensors Journal*, Vol. 19, Issue 11, June 2019.
12. Arduino Official Website. https://www.arduino.cc
13. Raspberry Pi Foundation. https://www.raspberrypi.org
14. MQ Gas Sensor Series Datasheet. https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-2.pdf
15. HX711 Load Cell Amplifier Datasheet. https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf
16. A. Verma, S. Agarwal, "IoT-Based Smart Gas Monitoring and Alerting System Using ESP8266", *International Journal of Computer Applications*, Vol. 179, No. 7, 2018.

17. B. Singh and S. Sharma, “Application of Random Forest in Health Monitoring Systems”, *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 6, Issue 4, April 2017.
18. A. Jain and V. Kumar, "Performance Evaluation of Linear Regression in Predictive Analysis", *International Journal of Computer Applications*, Vol. 113, No. 1, 2015.
19. G. Marquez et al., "A Machine Learning Approach for Predicting Gas Cylinder Weight", *Procedia Computer Science*, Vol. 170, pp. 657–662, 2020.
20. Python Software Foundation, “Scikit-learn: Machine Learning in Python.” https://scikit-learn.org