24EEE431

AI AND EDGE COMPUTING

END SEMESTER PROJECT

REPORT

SMART DAMAGE/LEAKAGE DETECTION SYSTEM

Team Members:

Aakash Sriram – CB.EN.U4ELC22001

Mohithaa K – CB.EN.U4ELC22029

Mukhil R – CB.EN.U4ELC22033

Pavitra A – CB.EN.U4ELC22040

1. Abstract

Oil and gas pipelines are essential infrastructures for fuel and gas transportation particularly in smart cities and industrial areas. However, these pipelines are prone to damage and leakage, which can lead to severe safety hazards and financial losses. The proposed system uses Machine Learning based gas detection and enables real-time monitoring and remote control of gas flow. The system uses gas sensors connected to a Raspberry Pi to detect leakages and classify gas types. Detected leaks and the type of gas are instantly alerted to the via a mobile application, enabling remote shut-off when a smoke is detected. This approach enhances the efficiency of pipeline operations and significantly improves safety in smart city infrastructures.

2. Introduction

Pipelines form the backbone of fuel and gas distribution in urban and industrial settings. However, undetected leaks can lead to catastrophic consequences, including explosions, fires, and exposure to toxic gases. Traditional inspection methods are often slow, reactive, and inefficient in preventing accidents. They often fail to detect leaks in real time, increasing the risk of large-scale damage, financial losses, and environmental harm. Seven gas sensors in this system can monitor gas levels at regular intervals, while a ML model, Random Forest Classifier analyzes patterns to differentiate between gas types and detect anomalies. Mobile platforms allow users to access real-time data, and remote-control mechanisms to enable immediate action in case of a leak. The objective of this project is to develop an intelligent, real-time gas leakage detection system that enhances safety and efficiency. By deploying multiple gas sensors, AI-driven classification, automated alerts, and remote-control features, the system will ensure better protection and smarter maintenance strategies. This solution will not only minimize hazards but also improve the reliability and sustainability of gas pipeline networks in smart cities.

3. Problem Statement

Smart cities often have communal LPG pipeline networks that require constant monitoring to prevent leaks. If undetected, gas leaks can result in serious accidents, financial losses, and environmental risks. Existing methods of leak detection rely heavily on manual inspections and outdated monitoring systems, which are slow and ineffective.

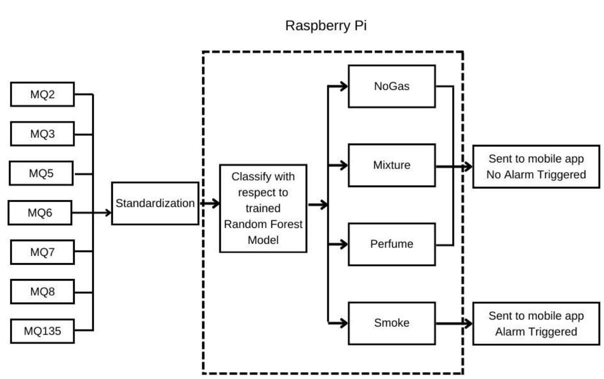
A technology-driven solution that enables real-time leak detection, predictive maintenance, and remote control is crucial for ensuring safety and efficiency in modern pipeline infrastructures.

4. Methodology

1. System Components and Architecture

The proposed system is designed for real-time application in city environments where pipelines are installed at 0.5 km intervals. It utilizes a Raspberry Pi as the central processing unit, connected to seven different gas sensors: MQ2, MQ3, MQ5, MQ6, MQ7, MQ8, and MQ135. These sensors continuously monitor gas concentration levels and classify detected gases into four categories: NoGas, Perfume, Smoke, and Mixture. The collected data is processed using Random Forest Classifier to identify potential leaks and notify users in real-time.

Block Diagram:



2. **Sensor Deployment and Communication:**

Gas sensors are installed at every 0.5 km segment of the pipeline. The sensors continuously transmit data via web socketing. A Raspberry Pi processes the received data and utilizes machine learning techniques of Random Forest Classifier to detect anomalies that may indicate a leakage. This real-time data processing ensures timely detection and helps in taking proactive measures.

3. **Communication Protocols used:**

A mobile application called IoT MQTT panel is connected to the Raspberry Pi 4 (Server) which communicates using Message Queueing Telemetry Transport (MQTT) protocol. It is a light-weight, publish subscribe messaging protocol. Eclipse Mosquitto is an open source MQTT broker that implements the MQTT protocol. It acts as the central server, managing the MQTT clients.

Enabling Mosquitto broker:

```
(aiedge) pi@raspberrypi:~/Downloads/aiedge $ mosquitto
1743223465: mosquitto version 2.0.11 starting
1743223465: Using default config.
1743223465: Starting in local only mode. Connections will only be possible from
clients running on this machine.
1743223465: Create a configuration file which defines a listener to allow remote
 access.
1743223465: For more details see https://mosquitto.org/documentation/authenticat
ion-methods/
1743223465: Opening ipv4 listen socket on port 1883.
```

Connection from server using Web Socket:

```
(aiedge) pi@raspberrypi:~/Downloads/aiedge $ python3 server.py
[SERVER] Listening on 192.168.120.33:5000...
[SERVER] Client connected from ('192.168.120.33', 52686)
```

Connecting to Server using Web Socket:

```
(aiedge) pi@raspberrypi:~/Downloads/aiedge $ python3 client.py
[CLIENT] Connected to server.
[CLIENT] Sending normal sensor values...
[CLIENT] Sent data: 748,339,323,473,466,578,428
[CLIENT] Sent data: 622,453,409,529,579,792,359
[CLIENT] Sent data: 509,474,559,494,783,506,373
[CLIENT] Sent data: 560,505,391,346,652,540,322
[CLIENT] Sent data: 754,407,498,443,525,719,518
```

Real-time Gas type Classification:

```
[SERVER] Detected: Smoke
[SERVER] Detected: Perfume
[SERVER] Detected: Mixture
[SERVER] Detected: Smoke
```

Smoke Simulation:

```
[CLIENT] Sent SMOKE data: 538,407,388,401,599,591,297
[CLIENT] Sent SMOKE data: 781,413,375,395,573,595,278
[CLIENT] Sent SMOKE data: 628,415,363,378,580,540,282
[CLIENT] Sent SMOKE data: 725,425,317,357,593,667,357
[CLIENT] Sent SMOKE data: 665,423,380,388,610,645,288
[CLIENT] STOP command received. Halting pipeline.
[CLIENT] Pipeline Closed. Waiting for RESET...
```

Gas Detected:

```
[SERVER] Detected: Smoke
[SERVER] Smoke detected, monitoring duration...
[SERVER] Detected: Smoke
[SERVER] Detected: Smoke
[SERVER] Detected: Smoke
[SERVER] Detected: Smoke
[SERVER] Smoke persisted for 9s! Triggering alarm...
[SERVER] Received MQTT message: ENABLE
[SERVER] Waiting for RESET command...
[SERVER] Waiting for RESET command...
[SERVER] Waiting for RESET command...
[SERVER] Waiting for RESET command...
[SERVER] Waiting for RESET command...
[SERVER] Waiting for RESET command...
[SERVER] Waiting for RESET command...
```

4. **Alert Mechanism & Control**

   If smoke is detected, a timer starts to monitor its persistence. If smoke persists for 9 seconds, an alarm is triggered, and a warning is sent via MQTT. The system shuts-off the pipeline preventing gas leakages. A RESET command is required to resume normal operation.

5. Solution Approach

1. **IoT-Based Monitoring System:**

The system is built around an advanced monitoring mechanism that integrates IoT-enabled gas sensors and AI-based leakage detection. The Raspberry Pi acts as the processing unit, collecting real-time data from sensors and analyzing it for potential leaks. The entire system is designed to ensure reliability, and response in case of emergencies.

2. **Real-Time Alert and Notification System:**

A mobile application is used to send real-time alerts to users, notifying them about detected leaks along with the exact time. This ensures that users can act quickly to mitigate risks. Additionally, a mobile app can resume the operations, once the problem is fixed, with a button in the mobile application.

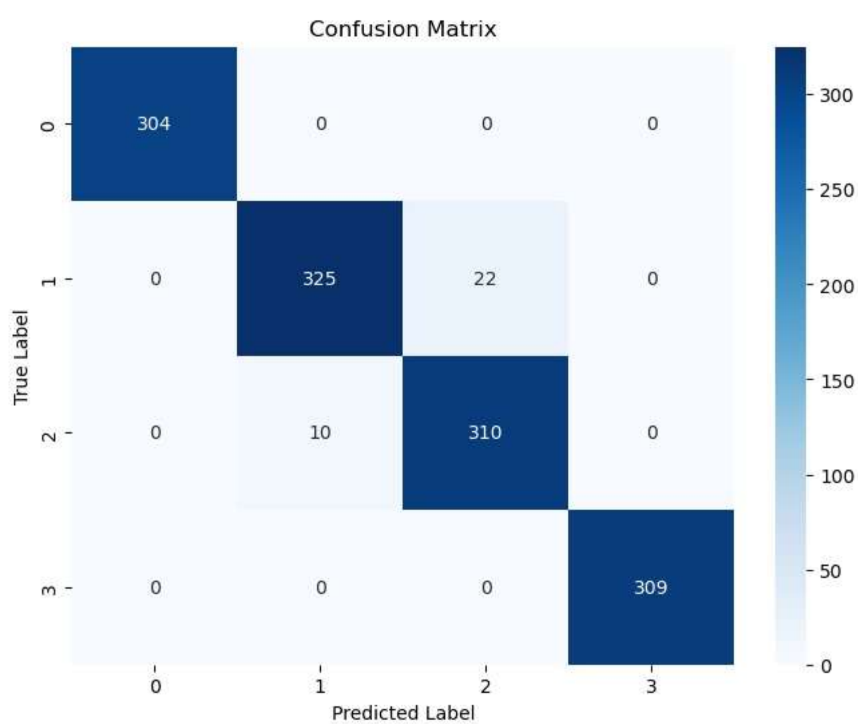
6. Result

The primary outcome of this system is the establishment of a real-time monitoring network for gas pipelines, ensuring continuous surveillance and leak detection. Automated alerts will provide instant notifications, enabling quick responses to minimize hazards. By incorporating remote control features, users can take immediate action to stop gas flow when necessary.

With emergency response integration, the system will improve response times to critical situations, minimizing the risk of large-scale incidents. Overall, the implementation of this system will significantly improve pipeline safety, operational efficiency, and energy distribution in smart city infrastructures.

The Machine Learning Model is evaluated by plotting a confusion matrix, and generating a classification report. The model has obtained an accuracy of 97.50%.

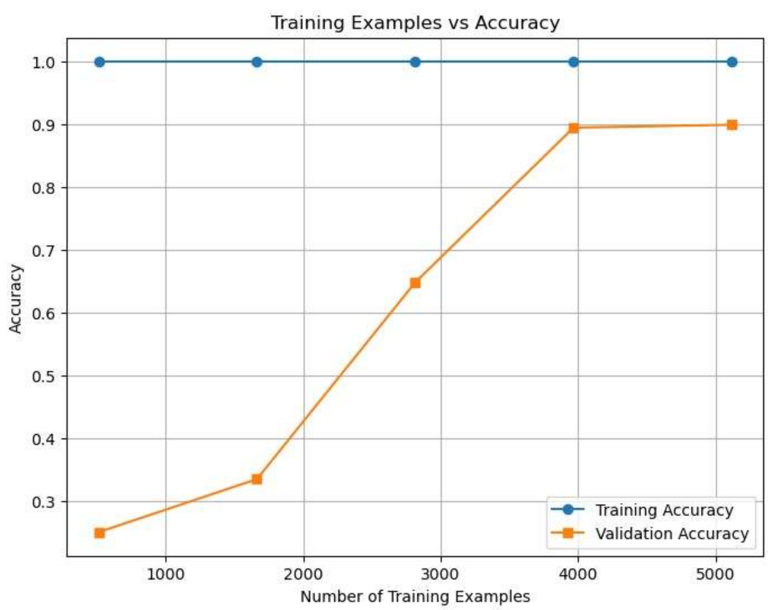
Confusion Matrix:



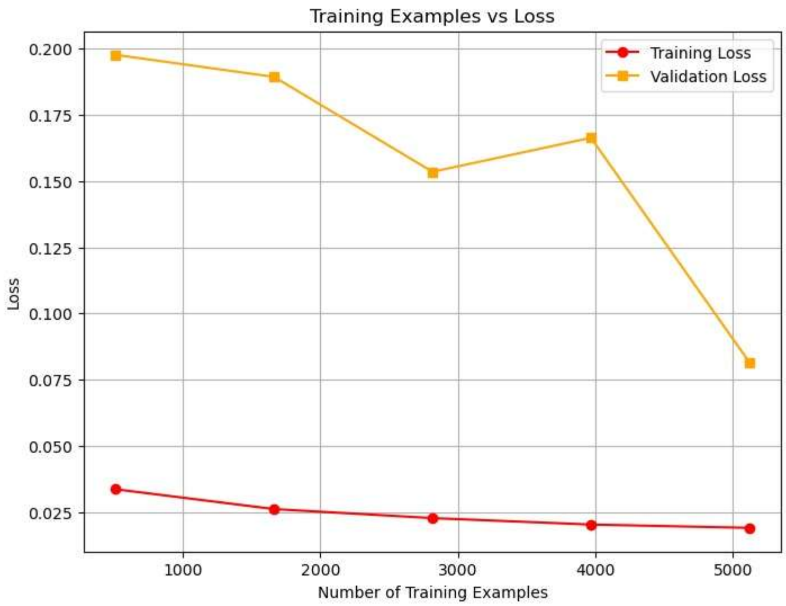
Classification Report:

```
Test Accuracy: 0.9750

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       304
           1       0.97      0.94      0.95       347
           2       0.93      0.97      0.95       320
           3       1.00      1.00      1.00       309

    accuracy                           0.97      1280
   macro avg       0.98      0.98      0.98      1280
weighted avg       0.98      0.97      0.98      1280
```

Accuracy vs Training Examples:



Loss vs Training Examples:



7. Conclusion

The proposed system integrates IoT and AI to create an advanced leakage detection and prevention mechanism for gas pipelines. By providing real-time monitoring, automated alerts, and remote-control features, it significantly enhances safety and reliability. Predictive analytics further ensures proactive maintenance, reducing risks and improving efficiency. The system's integration with emergency response teams and redundant safety mechanisms makes it a highly reliable solution for smart city infrastructures. Implementing this technology will lead to safer, more efficient pipeline operations, reducing hazards and ensuring better energy distribution.