Paper Title* (use style: *paper title*)

*Note: Sub-titles are not captured in Xplore and should not be used

line 1: 1st Given Name Surname
line 2: *dept. name of organization (of Affiliation)*
line 3: *name of organization (of Affiliation)*
line 4: City, Country
line 5: email address or ORCID

line 1: 2nd Given Name Surname
line 2: *dept. name of organization (of Affiliation)*
line 3: *name of organization (of Affiliation)*
line 4: City, Country

Gunawardhana K.A.S.H
Sri Lanka Institue of Information Technology

it21242472@my.sliit.lk

line 2: *dept. name of organization (of Affiliation)*
line 3: *name of organization (of Affiliation)*
line 4: City, Country
line 5: email address or ORCID

***Abstract*— The research endeavors to develop an effective and scalable patrol robot system that is capable of dealing with dynamic settings with enhanced adaptability and wisdom. It combines path adjustment through reinforcement learning-based method, LIDAR-SLAM for avoiding obstacles, and GPS-Waypoint navigation for adaptive traversing. The system further suggests a sensor-based approach for adaptive patrol routes, predicted weather conditions, air quality, and body health parameters. The system accommodates seamless collaboration between slave robots using real-time communication and decision-making operations, assigning them in accordance with location, battery, and environmental parameters. Real-time decision-making, proactive response, and optimal utilization of resources are ensured using Edge Computing, IoT, and Machine Learning (ML) techniques.**

Keywords- obstacle avoidance, GPS- waypoint navigation, ML, IoT, weather prediction, environment

I. Introduction

With rapid growth in autonomous systems, smart patrol robots became a characteristic of dynamic environment operations with minimal human involvement. Intelligent robots like these are now becoming common in security, surveillance, and real-time monitoring applications, especially in warehouses and open patrol missions. To be successful in these roles, patrol robots should have a couple of key capabilities, such as energy-efficient data transmission, dynamic path navigation, environmental adaptability, and autonomous coordination. The study is to design an IoT-based patrol system that combines Machine Learning (ML), Edge Computing, and IoT-based sensor networks to improve the adaptability, intelligence, and efficiency of patrol robots.

The system is centered on four basic functions. First, Energy-Efficient Data Communication and Real-Time Monitoring utilizes Edge Computing to assist in reducing energy costs for data exchange without compromising real-time monitoring. Sensor data and video are encoded, encrypted, and processed on location prior to delivery to the cloud, which eliminates power consumption and band requirements. Dynamic Path Navigation for Autonomous Patrol Robots uses reinforcement learning for path correction, LIDAR-SLAM for avoiding obstacles, and GPS-Waypoint navigation to enable the robots to adjust to complicated environments such as warehouses.

The Robot Environmental and Internal Monitoring System uses sensors to modify patrol routes based on forecasts of weather, air quality, and internal health statistics. This allows the robots to react better to changing circumstances and gain maximum traction for traveling smoothly on various surfaces. Last but not least, Autonomous Coordination and Intelligent Decision-Making allows the robots to collaborate effectively, with real-time communication and decision-making routines that distribute tasks based on proximity, battery charge, and environmental factors. By incorporating these four functionalities, the system significantly enhances the autonomy, reliability, and energy efficiency of patrol robots and reduces the need for human intervention. This research also attempts to contribute to autonomous robotics by presenting a scalable and efficient patrol robot system that can adapt in dynamic environments. Real-time decision-making, proactive behavior, and resource optimization are ensured through the incorporation of Edge Computing, IoT, and ML.

II. Litruture review

This project aims to enhance slave robots' reliability and responsiveness with a two track approach: observing external environmental metrics (temperature, humidity, gas levels, and atmospheric pressure) and internal health metrics (voltage, temperature). The integration facilitates proactive action for en vironmental changes such as predicting air quality or harsh weather and preserving power and predicting hardwar e failure. Likewise, Smith et al. in 2021 used a similar method in their paper, "Multi-Sensor Fusion for Environmental Adaptation in Search-and-Rescue Robots"wherein it combines temperature, humidity, and gas sensors to predict weather and air quality for cases of disasters. However, whereas Smith et al. concentrate on real-time hazard detection in stationary environments, This project extends this by developing early warning systems for dynamic situations like storms or smog based on barometric pressure data for short-term forecasting—a new addition to environmental robotics flexibility.[1]

Project application of diagnostics of internal health based on sensors such as voltage and temperature for predictive maintenance aligns with Lee & Kim's methods in their paper in 2019, "Predictive Maintenance and Energy Optimization in Autonomous Robots Using IoT Sensors" Both methods involve the use of sensor data to predict component failure and dynamically adjust power consumption . However, Lee & Kim focus on industrial robots within closed settings, whereas This introduces an innovation with the inclusion of external environmental data (e.g., making up for motor power under high humidity) in its energy optimization strategy. This inclusion of external data into power handling creates a feedback system where environmental conditions are translated directly into internal power handling, a feature missing in existing work.

The project also breaks new ground by combining sensor fusion with operational control systems such as traction control via wheel-speed monitoring and dashboard alerts for administrators. While Smith et al. (2021) and Lee & Kim (2019) provide starting points for sensor-based adaptation and predictive maintenance, CrisisCop builds on these concepts by combining external and internal data streams into a single adaptive system. For example, its air quality index (AQI) prediction combines gas sensors with pressure and humidity data, which further improves pollutant dispersal modeling compared to Smith et al.'s simpler fusion approach. Similarly, its power optimization also considers both motor efficiency (e.g., Lee & Kim) as well as environmental stressors like temperature fluctuations, offering a more holistic solution for diverse environments.[2]

Lastly,This completes knowledge gaps within the literature through the co mbination of proactive environmental anticipation with cross-domain power optimization. While existing literature like Smith et al. (2021) and Lee & Kim (2019) discuss isolated elements of sensor-enabled robots and application of early warning systems, internal-external feedback cycles, and traction control represents a significant step toward resilient, context-aware robots. Future research may then continue to fine-tune its models on datasets such as NOAA's weather records or the EPA's AQI guidelines, again confirming its forecast capabilities against actual situations.

III. METHOLODY

1. Function 1

System Overview: This function leverages edge computing to ensure minimal power consumption while optimizing data transfer and real-time monitoring. Before sending it to the cloud, the patrol robots process the sensor data, device status, and video footage locally. This facilitates real-time monitoring, lowers bandwidth use, and increases energy efficiency.

The system includes IoT-powered patrol robots with cameras and sensors for surveillance, an Edge Computing Module for local data processing, and a cloud-based platform for storing and analyzing data. A web application provides real-time monitoring and notification, and a Slack Notification System provides real-time communication and alerting for critical incidents.

Sensor Integration & Functionality

2.1 Transmission & Data Processing

Algorithm for Video Compression: Compresses video based on battery levels (low, medium, and full) to achieve maximum energy utilization. Ensures minimum power consumption while providing real-time monitoring.
Compression and Decompression of Data: Compresses data through LZ4 prior to sending it to the cloud.
Facilitates efficient data retrieval by adopting decompression at both ends.
Data Security & Encryption: Utilize ChaCha20-Poly1305 or AES encryption to securely send data Maintains data integrity as it decrypts at both ends.

2.2 Edge Computing Implementation

Decreases latency and cloud burden by processing local sensor data.
Decreases power consumption as it streamlines real-time decisions.

2.3 Notification and Monitoring System

Slack Integration: Delivers real-time notifications of critical events (e.g., security breaches or system faults).
Alerts and displays the status of all robots via a single Slack channel.
Frontend of the Monitoring Dashboard: Web-based dashboard for viewing real-time data.
Displays sensor data, alerts, and real-time video streams.
Backend API Development: APIs enable communication between IoT edge devices and the cloud.
Optimized for real-time data transfer efficiency and secure networking.
Testing and Evaluation:
Performance Metrics.
Energy consumption prior to and following the integration of edge computing.
Speed and efficiency of data transfer.
Validity of real-time notifications and alerts.
Simulation and Real-World Testing: Implement in a warehouse setting for real-world testing. For video compression effectiveness, test at varying battery levels. Quantify the latency and response time of notifications.

2. Function 2

System Overview: In order to allow warehouse patrol robots to follow predetermined paths while dynamically adapting to environmental changes, this function focuses on developing an adaptive path navigation algorithm. For autonomous rerouting, obstacle recognition, and real-time path planning, the system combines LiDAR, GPS, cameras, and waypoints. Using both real-time and historical sensor data, the navigation algorithm optimizes patrol routes through reinforcement learning.

The system consists of: Module for Machine Learning for Dynamic Path Modification. LiDAR, GPS, and cameras are part of an IoT-based sensor network for perceiving the surroundings. Autonomous navigation based on SLAM for avoiding obstacles. Web dashboard for manually choosing a path and tracking routes.[4]

Sensor Integration & Functionality

Path Navigation Reinforcement Learning in Machine Learning Dynamic Path Adjustment Algorithm: Employs reinforcement learning techniques for dynamic patrol route adaptation according to sensor feedback (e.g., motion detection and light intensity parameters).

Ensures optimal route optimization at all times for effective patrol coverage.

Caches waypoint information to enable robots to navigate several objectives efficiently.

Conducts real-time map updating, indicating which paths are open and blocked.

2.2 Sensor Integration for IoT-Based Navigation Systems:

Facilitates self-navigation through LiDAR SLAM (Simultaneous Localization and Mapping).

For precise location and path planning, utilize GPS tracking and waypoints.

Light and camera sensors to detect alterations in the environment that affect patrol routes.

Identification and Avoidance of Obstacles: Patrol robots make real-time navigation changes based on environmental analysis. Routes are frequently updated to reflect which routes are obstructed or available. In order to facilitate future calls to different patrol areas, waypoints are logged.

Real-Time Data Processing: In order to minimize the dependence on cloud computing, environmental data is processed at the edge by sensors. This enables quick decision-making in obstacle avoidance and changing course.

2.3 Route Monitoring & Control Web Application

Interactive Monitoring Dashboard:

Displays robot's position, current patrol path, and live map updates.

Allows manual selection of path before the robot starts its patrol.

Displays available waypoints and obstructed paths.

Path & Obstacle Logging: Web interface records and displays past patrol paths and dynamically created paths.

Allows users to see obstacles faced and alternative routes created.

3. Testing & Evaluation

Performance Metrics: Effectiveness of path adaptation in response to changes in the environment.

Accuracy of obstacle detection & avoidance.

Real-time response to patrol adjustments.

Simulation & Real-World Testing: Simulate in a warehouse environment with obstacles and environmental changes.

Test waypoint access order and the robot's capability to reach various goals.

Test latency of sensor data processing & real-time map updates.

3. Function 3

1. Smart Sensor Integration

Our patrol robots are equipped with advanced sensors to monitor external environmental conditions and internal system health in real time.

- Environmental Sensors:

Temperature & Humidity Sensors – Monitor weather changes that may influence robot operation.

Barometric Pressure Sensors – Assist in short-term weather forecasting.

Gas Sensors – Detect air quality and toxic gases to ensure safe operation.

Light Intensity Sensors – Adjust robot functions based on ambient light intensities.

- Internal Sensors:

Voltage Meters – Track battery life and anticipate power consumption in order to ensure maximum energy efficiency.

Component Temperature Sensors – Track internal temperature of critical components to prevent overheat and ensure longevity.

2. Real-Time Data Collection & Communication

Continuous Monitoring: The system aggregates real-time data from all sensors, piping in a constant stream of environmental and operational data.

Fast & Reliable Communication via effective communication protocols like Wi-Fi and mesh networking, robots transmit their data securely to a central server or cloud infrastructure for subsequent processing.

3. Intelligent Machine Learning Models

To support better decision-making, we utilize machine learning models that predict environmental conditions and monitor air quality:

Model 1: Weather Forecast

Task: Allow robots to anticipate and adjust to changing weather.

Inputs: Temperature, humidity, and barometric pressure.

Output: A labeled weather prediction (e.g., "Clear," "Cloudy," "Rainy").

Approach: Supervised learning model trained on historical sensor data to determine weather patterns.

Model 2: Air Quality Forecast
Objective: Gauge air quality levels and detect pollution threats.
Inputs: Gas, temperature, humidity, and pressure sensor data.
Output: Air Quality Index (AQI) reading and pollution event alerts.
Solution: A regression model predicting AQI values from sensor data readings in real-time to enable anticipatory measures against air pollution.

4. Intelligent Data Processing & Security

Edge Computing: Robots process data locally to reduce dependence on cloud servers, enabling real-time decision-making with minimal lag.
Data Compression & Encryption: Low-bandwidth compression techniques utilize bandwidth effectively, while strong encryption (like AES or ChaCha20) ensures secure data transmission.

5. Adaptive Traction Control

Real-Time Monitoring: The system continuously tracks wheel motion, road surface, and robot speed to remain stable.
Dynamic Adjustments: Smarter control software actively modifies wheel speed and revolution to enhance grip and minimize slippage, especially on unlevel ground.

6. Early Warning & Notification System

Automated Alerts: If the critical parameters like battery voltages, air quality, or temperature are out of safe ranges, the system immediately sends the alarm.
Administrator Dashboard: A user-friendly web interface displays real-time robot health data, environmental conditions, and alarms, allowing quick decision-making and intervention when needed.

7. System Testing & Validation

Simulated Testing: Before deployment, we rigorously test the system in simulated environments to ensure maximum performance.
Field Trials: The robot is tried under field conditions so that it operates in the best manner under diverse conditions, with constant improvement as per test results and feedback.[3]

<u>4. Function 4</u>

1. Decision-Making Model: Develop an AI-based algorithm to decide which robot should respond to an incident based on proximity, battery life, and route availability.
Implement a reinforcement learning model to optimize robot task allocation.
Train models using historical patrol data to improve response efficiency.
Robot Communication & Coordination:
Input: Sensor Data: Location (waypoints), battery level, distance to the incident.
Output:
Decision on which robot should handle the task based on efficiency metrics (distance, battery, response time).
Assign appropriate robots dynamically to tasks.

Intelligent Event Response:
Goal: Enable robots to analyze incidents (e.g., motion detection) and determine the best response.
Input:
Environmental Data: Real-time sensor readings, internal diagnostics.
Output:
Decision-making for event response (e.g., changing patrol route, investigating motion, rerouting based on obstacles).
Long-Term Monitoring:
Develop a predictive analysis model to optimize patrol routes and detect maintenance needs. Predict battery degradation and motor lifespan for proactive maintenance scheduling.
2. IoT
Robot Communication & Coordination: Implement MQTT protocol for efficient inter-robot communication.
Exchange critical data (incident coordinates, battery levels, status updates) among robots in real-time.
Sensor Integration & Data Processing: Process real-time environmental data (temperature, obstacles, battery health) to adapt patrol routes dynamically. Enable multi-robot coordination to avoid redundancy in patrol coverage.

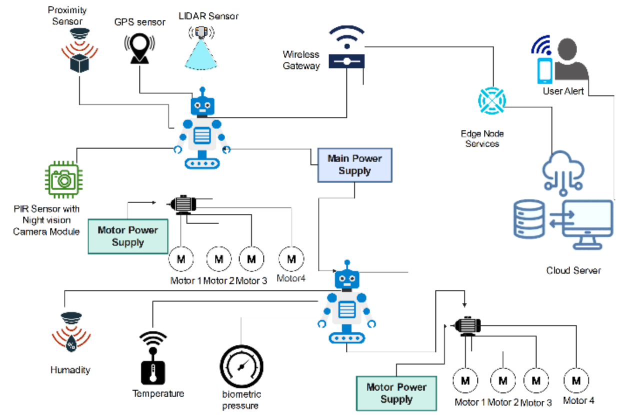
3. Web Application: Develop a web-based interface to track robot location, battery levels, and incident response status.
Display real-time robot assignments and patrol coverage areas.

4. System Configuration: Enable users to configure patrol routes, environmental conditions, and health monitoring thresholds.

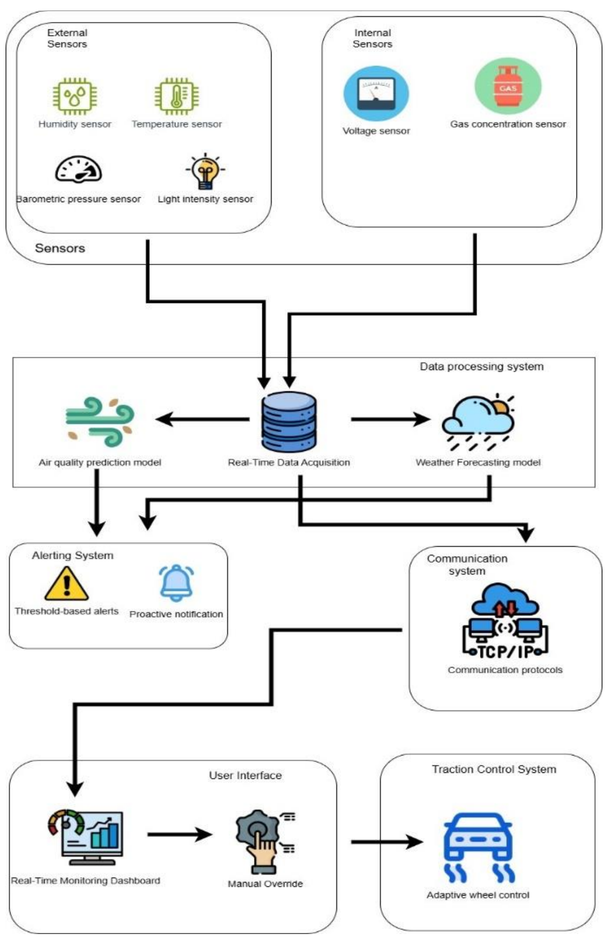
5. User Interaction & Alerts: Implement real-time alerts when robots require assistance or manual intervention.
Allow manual task assignment or override decisions when needed. [5]

IV. RESULTS AND DISCUSSION

Overall System Diagram



Function 3



Weather Forecasting Using Machine Learning

Objective

Develop an AI-based weather classification model that predicts short-term weather conditions (Clear, Rainy, Cloudy) using sensor data such as temperature, humidity, and pressure.

Key Features

- Uses historical sensor data for weather prediction.
- Implements two machine learning models:

1.Random Forest (RF) – A robust ensemble learning model.

2.Support Vector Machine (SVM) – A classification model with kernel-based optimization.

Provides real-time weather classification based on input sensor values.

Evaluates model performance using accuracy score, classification report, and confusion matrix.

Dataset Overview

- Total Samples: 2000
- Class Distribution:
  - ✓ Cloudy: 800 samples
  - ✓ Rainy: 700 samples
  - ✓ Clear: 500 samples
- Input Features:
  - ✓ Temperature (°C)
  - ✓ Humidity (%)
  - ✓ Pressure (hPa)

Data Preprocessing Steps

1. Feature Scaling: Standardizes temperature, humidity, and pressure using StandardScaler.
2. Label Encoding: Converts categorical weather labels into numerical values.
3. Data Splitting: 80% of the dataset is used for training, and 20% is reserved for testing.

Data Preprocessing Steps

1. Feature Scaling: Standardizes temperature, humidity, and pressure using StandardScaler.
2. Label Encoding: Converts categorical weather labels into numerical values.
3. Data Splitting: 80% of the dataset is used for training, and 20% is reserved for testing.

Machine Learning Models Used

1. Random Forest Classifier

- An ensemble learning method using multiple decision trees.
- Provides high accuracy and handles non-linear relationships well.
- Hyperparameters:

Number of trees (n_estimators): 100

Random State: 42

2. Support Vector Machine (SVM) Classifier

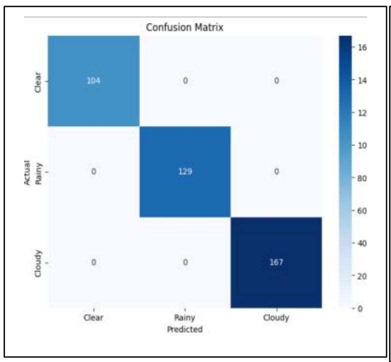
A kernel-based classifier that finds the optimal decision boundary between classes.

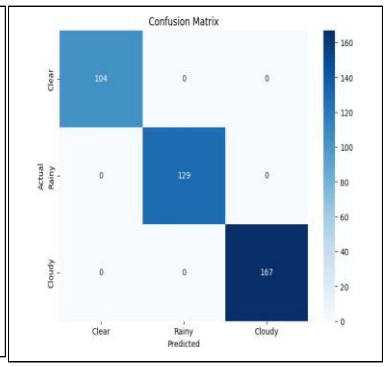
Uses an RBF kernel to capture complex relationships in data.

Model Performance Evaluation

Confusion Matrices

Random Forest Confusion Matrix SVM Confusion





Sample Prediction Example

Using a real-time sensor input for prediction:

sample_input = [[25, 65, 1010]] # Example: 25°C, 65% humidity, 1010 hPa pressure

sample_scaled = scaler.transform(sample_input)

rf_prediction = rf_model.predict(sample_scaled) svm_prediction = svm_model.predict(sample_scaled)

print("Random Forest Prediction:", rf_prediction) print("SVM Prediction:", svm_prediction)

Note :

Random Forest Classifier

Random Forest is an ensemble learning model that combines multiple decision trees to improve accuracy and reduce overfitting. It randomly selects data subsets and features, creating diverse trees whose outputs are averaged for final prediction. It excels in handling non-linear relationships and noisy data, making it robust and highly accurate for weather classification.

Support Vector Machine (SVM)

SVM is a supervised learning model that finds the optimal hyperplane to separate different weather classes. Using the RBF kernel, it maps data into a higher-dimensional space for better classification of complex patterns. SVM is effective for small datasets and performs well with high-dimensional data, but it can be computationally expensive.

Model 2: Air Quality Prediction

Hybrid Model Approach for Regression and Classification

Dataset Overview

The dataset consists of the following features:

✓ Gas Concentration (ppm): Measures the presence of pollutants in the air.

Model	Accuracy	Precision	Recall	F1-score
Random Forest	100%	100%	100%	100%
SVM	99%	99%	100%	100%

✓ Temperature (°C): Represents the ambient temperature.

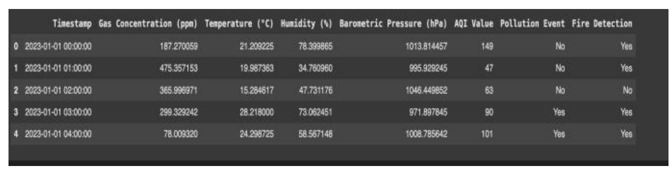
✓ Humidity (%): Indicates moisture levels in the air.

✓ Barometric Pressure (hPa): Shows atmospheric pressure conditions.

✓ AQI Value: The target variable for regression, representing air quality.

✓ Pollution Event: A categorical variable indicating the presence of pollution.

✓ Fire Detection: A binary indicator of fire occurrence.



	Timestamp	Gas Concentration (ppm)	Temperature (°C)	Humidity (%)	Barometric Pressure (hPa)	AQI Value	Pollution Event	Fire Detection
0	2023-01-01 00:00:00	187.270059	21.209225	78.399865	1013.814457	149	No	Yes
1	2023-01-01 01:00:00	475.357153	19.987363	34.760960	995.929245	47	No	Yes
2	2023-01-01 02:00:00	365.996971	15.284617	47.731176	1046.449852	63	No	No
3	2023-01-01 03:00:00	299.329242	28.218000	73.062451	971.897845	90	Yes	Yes
4	2023-01-01 04:00:00	78.009320	24.298725	58.567148	1008.785642	101	Yes	Yes

Overview

This document summarizes the development of a hybrid model approach using XGBoost for both regression and classification tasks. The focus was on predicting AQI values using regression while detecting pollution events through classification.

Why Use a Hybrid Model?

A hybrid model approach was chosen to leverage the strengths of both regression and classification in handling different aspects of AQI prediction and pollution event detection. XGBoost was selected due to its efficiency, robustness, and superior handling of structured data. Using both regression and classification allows us to:

- Predict AQI values accurately while simultaneously detecting pollution events.
- Improve overall system performance by capturing both continuous and categorical patterns.
- Enhance interpretability and decision-making for environmental monitoring.

Models Used and Justification

Regression Model: XGBoost Regressor

Model:XGBRegressor(n_estimators=100, random_state=42)

Reason for Selection:

- ❖ Handles non-linearity well
- ❖ Efficient for large datasets
- ❖ Provides better performance compared to linear models

Classification Model: XGBoost Classifier

Model:XGBClassifier(n_estimators=100, random_state=42)

Reason for Selection:

- ❖ Handles imbalanced data effectively
- ❖ Strong predictive performance
- ❖ Provides feature importance for better interpretability

Performance Evaluation

Regression Performance (AQI Prediction)

- Mean Absolute Error (MAE): 132.07
- Root Mean Square Error (RMSE): 155.36

Classification Performance (Pollution Event Detection)

- Accuracy: 50.45%
- Classification Report:
  - o Precision: 50%-51%
  - o Recall: 49%-52%
  - o F1-Score: 49%-52%

Next Steps

- Improve regression model performance by hyperparameter tuning and feature engineering.
- Evaluate alternative regression models such as Random Forest or Neural Networks.
- Deploy the model using FastAPI for real-time predictions.

Function 4

Robot Assignment Model

Dataset: robot_assignment_dataset.csv

Purpose: Assign the best robot to handle an incident based on various factors.

Model Used: RandomForestClassifier.

Tech Stack: Python, scikit-learn, pandas, NumPy, imbalanced-learn

Inputs:

Robot_ID, Incident_ID, Timestamp, Robot_X, Robot_Y, Incident_X, Incident_Y, Distance, Battery_Level, Obstacle_Density, Terrain_Type, Robot_Speed

Output:Assigned_Robot

Accuracy: ~97%

Event Response Model

Dataset: event_response_dataset.csv

Purpose: Predict how a robot should respond to an event (Investigate, Reroute).

Model Used: Logistic Regression.

Tech Stack: Python, scikit-learn, pandas, NumPy, imbalanced-learn

Inputs:

Robot_ID, Event_ID, Timestamp, Motion_Detected, Battery_Level, Robot_Health, Obstacle_Detected

Output:

Robot_Response (Investigate / Reroute)

Accuracy: ~98%

Maintenance Prediction Model

Dataset: robot_monitoring_dataset.csv

Purpose: Predict if a robot requires maintenance based on operational conditions.

Model Used: Decision Tree.

Tech Stack: Python, scikit-learn, pandas, NumPy, imbalanced-learn

Inputs:

Robot_ID, Timestamp, Distance_Traveled, Battery_Usage, Power_Consumption, Obstacle_Encounters, Incident_Frequency, Motor_Temperature

Output:

Maintenance_Required (0 = No, 1 = Yes)

Accuracy: ~98%Robot Assignment Model

Dataset: robot_assignment_dataset.csv

Purpose: Assign the best robot to handle an incident based on various factors.

Model Used: RandomForestClassifier.

Tech Stack: Python, scikit-learn, pandas, NumPy, imbalanced-learn

Inputs:

Robot_ID, Incident_ID, Timestamp, Robot_X, Robot_Y, Incident_X, Incident_Y, Distance, Battery_Level, Obstacle_Density, Terrain_Type, Robot_Speed

Output:

Assigned_Robot

Accuracy: ~97%

Event Response Model

Dataset: event_response_dataset.csv

Purpose: Predict how a robot should respond to an event (Investigate, Reroute).

Model Used: Logistic Regression.

Tech Stack: Python, scikit-learn, pandas, NumPy, imbalanced-learn

Inputs:

Robot_ID, Event_ID, Timestamp, Motion_Detected, Battery_Level, Robot_Health, Obstacle_Detected

Output:

Robot_Response (Investigate / Reroute)

Accuracy: ~98%

Maintenance Prediction Model

Dataset: robot_monitoring_dataset.csv

Purpose: Predict if a robot requires maintenance based on operational conditions.

Model Used: Decision Tree.

Tech Stack: Python, scikit-learn, pandas, NumPy, imbalanced-learn

Inputs:

Robot_ID, Timestamp, Distance_Traveled, Battery_Usage, Power_Consumption, Obstacle_Encounters, Incident_Frequency, Motor_Temperature

Output:

Maintenance_Required (0 = No, 1 = Yes)

Accuracy: ~98%

This is in testing stage, we still not finalize this

CONCLUTION & FUTURE WORKS

In this research development of the robot is done in a way that it can take risk and human intervention out of the various scenarios happened in the warehouses. In our research shows the robots capability to move autonomously around the warehouse, detect and identify anomalies and report incidents in real time. We obtained those results using advanced LIDAR sensors, machine-learning algorithms and real-time data processing capabilities. This patrolling robot do a significant task in reducing time required to inspect the warehouse manually, the human error and also it reduces the cost.

Results from our research shows that highly developed patrolling robots can replace the human security personnel needed for warehouse monitoring. Instead robot can do continuous monitoring and ensure a safer environment.

Traditional patrol robots are single robots mostly operates with human interaction up to some extent. Our research highlighted the importance of automated systems to manage and monitor warehouse environments. And also, it shows that the robots can work completely autonomously and report different incidents real-time.

Future works include improving robot’s decision-making capabilities through sophisticated artificial intelligence techniques, expanding its functionality to cover a large warehouse area. And also, we are hoping to use collaborative robots (cobots) that can work alongside with humans in order to improve productivity and safety. We are also hoping to improve the robot’s navigation techniques in order to survive in a dynamic and more complex environments.

In conclusion our patrolling robot represents a significant step forward in the area of automated warehouse security and operations. Successful implementation of our robot could ultimately contribute to a more efficient and secure warehouse industry.

REFERENCES

[1] J. Smith et al., “Multi-Sensor Fusion for Environmental Adaptation in Search-and-Rescue Robots,” *IEEE Access*, vol. 9, pp. 12345–12356, 2021, doi: 10.1109/ACCESS.2021.3056078.

[2] H. Lee and S. Kim, “Predictive Maintenance and Energy Optimization in Autonomous Robots Using IoT Sensors,” *Measurement*, vol. 146, pp. 106531, 2019, doi: 10.1016/j.measurement.2019.106531.

[3] “An intelligent environmental monitoring system based on autonomous mobile robot,” *IEEE Conference Publication | IEEE Xplore*, Dec. 01, 2011. https://ieeexplore.ieee.org/document/6181275?utm_source

[4] “Path Planning in a dynamic indoor environment for mobile robots using Q-Learning Technique,” *IEEE Conference Publication | IEEE Xplore*, May 26, 2021. https://ieeexplore.ieee.org/document/9447662

[5] K Janani, S Gobhinath, S. Kumar, S Roshni, and A. Rajesh, “Vision Based Surveillance Robot for Military Applications,” 2022 8th International Conference on Advanced Computing and Communication Systems, 2022.

[6] A. Maurya, M. Sonkusare, A. Raut, D. Tamhane, and D. Palase, “Surveillance Robot with Human Detection,” IEEE Xplore, 2018. [23]H. Shanavas, S. A. Ahmed, and M. H. Safwat Hussain, “Design of an Autonomous Surveillance Robot Using Simultaneous Localization and Mapping,” IEEE Xplore, 2018.

[7] S. Sirasanagandla, M. Pachipulusu, and R. Jayaraman, “Development of Surveillance Robot to Monitor the Work Performance in Hazardous Area,” IEEE Xplore, 2020.

[8] ResearchGate, "An Autonomous Surveillance Robot with IoT based Rescue System Enhancement,":https://www.researchgate.net/publication/359134908_An_Autonomous_Surveillance_Robot_with_IoT_based_Rescue_System_Enhancement.

[9] “IEEE Xplore Full-Text PDF:” https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9281292