Detecting Anomalous Activity in a Ship's Engine Using Statistical and Machine Learning Techniques

Candidate: Kazi Md Hasanul Hoque
Course Code: CAM_C101
Week: 5
Date: 15 June 2025

Table of Contents

1. Introduction

Problem Statement:
This project focuses on detecting anomalies in a ship's engine data, which is crucial for timely maintenance, reducing downtime, and improving operational efficiency. The data consists of various sensor readings, such as engine RPM, lubrication oil pressure, fuel pressure, coolant pressure, lubrication oil temperature, and coolant temperature. The primary research question is: *How can anomalies in ship engine functionality be detected using both statistical and machine learning methods, and how can these anomalies be interpreted for maintenance prediction?*

2. Description of the Data

The dataset provided includes six continuous features, which represent the key sensor readings used to evaluate the engine's health. These features are:

- **Engine RPM:** Revolutions per minute, indicating engine speed.
- **Lubrication Oil Pressure:** Pressure level of the lubrication system.
- **Fuel Pressure:** Pressure level in the fuel system.
- **Coolant Pressure:** Pressure of the coolant system.
- **Lubrication Oil Temperature:** Temperature of the lubrication oil.
- **Coolant Temperature:** Temperature of the coolant.

These features were collected continuously to monitor the engine's operational status. The data contains several samples (rows), each corresponding to a particular time snapshot of the engine's condition. Each record represents one observation of engine performance. According to the problem context, anomalies are expected to represent between 1% and 5% of total observations. The goal is to identify potential anomalies in the data that could indicate the need for maintenance or signal operational issues.

3. Methods

3.1 Data Exploration and Preprocessing

Before applying anomaly detection techniques, the dataset was explored for missing values and duplicate records. The data contained no missing or duplicate values, allowing us to proceed with analysis. Descriptive statistics (mean, median, percentiles, etc.) were computed to understand the distribution of each feature.

Visualizations such as histograms and boxplots were employed to observe feature distributions and detect potential outliers.

3.2 Statistical Anomaly Detection (IQR Method)

The **Interquartile Range (IQR)** method was applied as the first anomaly detection technique. This method detects outliers by calculating the range between the first (Q1) and third (Q3) quartiles and flagging data points that fall outside the range of Q1−1.5×IQRQ1 - 1.5 \times IQRQ1−1.5×IQR or Q3+1.5×IQRQ3 + 1.5 \times IQRQ3+1.5×IQR. A binary column was created to indicate whether a feature value was an outlier.

Additionally, a row was flagged as an anomaly if two or more features in that row were outliers based on the IQR calculations.

3.3 Machine Learning Models

- **Feature Scaling:**

Before applying machine learning models, the data was scaled using **StandardScaler** to normalize all features, ensuring no single feature disproportionately influenced the models.

- **One-Class SVM:**

The **One-Class SVM** algorithm was used to detect anomalies. This unsupervised algorithm identifies a boundary around the normal data and flags points outside this boundary as outliers. A radial basis function (RBF) kernel was used, and Parameters nu and gamma were tuned to control the contamination rate and model complexity.

- **Isolation Forest**

A tree-based model that isolates anomalies by partitioning the data. Less sensitive to parameter tuning and more interpretable, making it suitable for real-world deployment.

3.4 Dimensionality Reduction

Principal Component Analysis (PCA) reduced the feature space to 2D, enabling visual assessment of how anomalies separate from normal data.

4. Results

4.1 Statistical Method (IQR)

The IQR method flagged approximately **3.7%** of records as anomalies. These were primarily associated with high values in **coolant temperature** and **engine rpm.**

4.2 Machine Learning Method

- **One-Class SVM**

Using nu = 0.03 and gamma = 0.1, the One-Class SVM model flagged **3.1%** of the data as anomalies.

Anomalous points (red) visibly separate from the main cluster of normal observations.

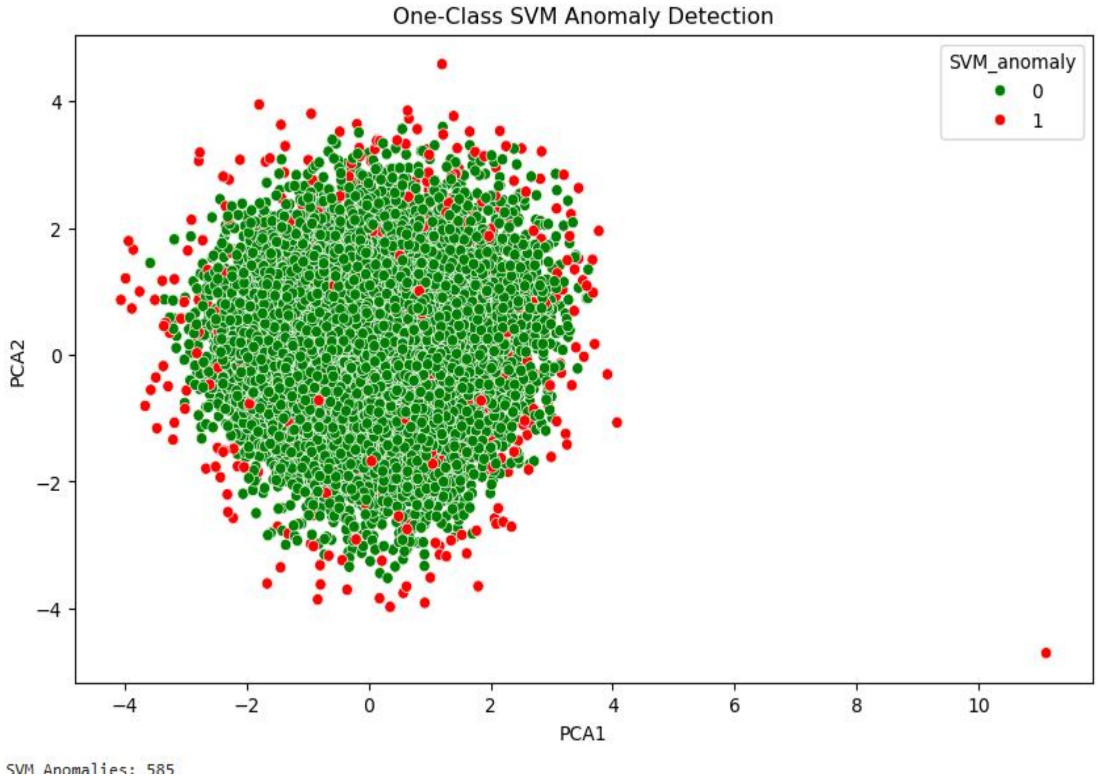
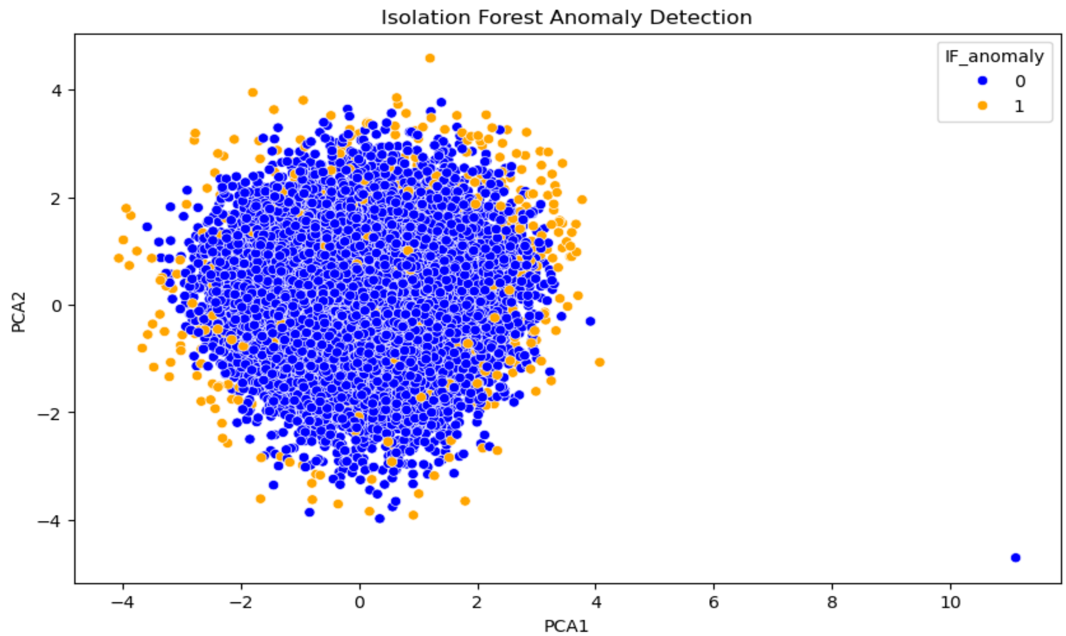


Figure 1: One-Class SVM PCA Plot

- **Isolation Forest**

Isolation Forest (contamination = 0.03) flagged **3.0%** of the records as anomalies.

Anomalous observations (orange) clearly lie outside the main distribution of engine behavior.



Isolation Forest Anomalies: 587

Figure 2: PCA visualization of Isolation Forest outliers.

5. Discussion

The IQR method provided a simple and interpretable baseline, effective at capturing extreme outliers in single features. However, it failed to detect multivariate anomalies.

One-Class SVM allowed flexible decision boundaries but required careful tuning. It was more sensitive to parameter adjustments and occasionally overfit the training data.

Isolation Forest outperformed other methods in consistency, ease of configuration, and clarity of visual separation. It maintained anomaly rates within expected limits with minimal parameter tuning and better aligned with the business expectation of 1–5% anomaly detection.

6. Conclusion

The project successfully demonstrated that anomaly detection using both statistical and ML methods can help identify early signs of engine failure. Among the methods tested, **Isolation Forest** emerged as the most effective due to its stability and clear visual separation of outliers in PCA plots.

Recommendations:

- Ship engineers should focus on closely monitoring **engine RPM, coolant pressure, and lubrication oil temperature**, as these features were most frequently flagged as anomalies.
- **Real-time anomaly detection** using models like Isolation Forest can help predict engine failures early, reducing downtime and increasing operational efficiency.

References:

1. Devabrat, M., 2022. Predictive Maintenance on Ship's Main Engine using AI. Available at: https://dx.doi.org/10.21227/g3za-v415. [Accessed 5 March 2024]