Fuel Leakage Detection in A400M Aircraft

AIRBUS

Capstone Group 4
10 July 2024

"I hereby certify that this report and the accompanying presentation is my own original work in its entirety, unless where indicated and referenced."

Katherine Bade | Cecilia Crepaldi | Pablo Ferraro | Andres Ferrer | Constantin Wiederin

Index

1. Executive summary

https://github.com/badek23/AirbusFuelLeaks

This report outlines a comprehensive approach to predicting and detecting fuel leakages in aircraft for Airbus, focusing on enhancing safety, reducing maintenance costs, and improving operational efficiency.

Current fuel leak detection in the industry often relies on monitoring fuel volume within tanks. The lack of precision inherent in this approach can make it difficult to identify leaks. The project's objective is to develop a data-driven automatic fuel leak detection model to mitigate these issues and enhance aircraft availability while reducing maintenance costs.

Fuel leak detection is currently performed at different flight phases:

- Before the flight, initial fuel calculations are made.
- During the flight, regular checks are performed to monitor fuel usage and detect abnormalities.
- After the flight, post-flight checks ensure there are no discrepancies in fuel usage data.

Fuel leaks impose significant financial burdens, including Aircraft on Ground (AOG) costs, maintenance expenses, and disruption costs. This report estimates that early detection and prevention of fuel leaks can save approximately $34,405 per aircraft monthly. These savings are derived from reducing AOG incidents and lowering maintenance costs.

We utilized historical datasets containing time-series data from aircraft sensors. We performed rigorous data cleaning to handle null values and focused our analysis on the cruise phase of flights, where conditions are most stable for detecting fuel inconsistencies.

Three main models were considered for anomaly detection: Autoencoders, Isolation Forests and XGBoost. The Autoencoder model, which compresses and reconstructs data to identify anomalies, showed better performance metrics compared to the Isolation Forest and XGboost models. The final Autoencoder model achieved an accuracy of 94.24%, recall of 86.95%, precision of 99.21%, and an F1 score of 92.67.

We propose this model be integrated into a user-friendly dashboard accessible to both aircraft crew and ground staff. Initially, the dashboard will be deployed to pilots on high-risk routes and older aircraft, followed by a fleet-wide rollout after user feedback and with continuous improvement.

Implementing this advanced fuel leak detection system will enhance Airbus's operational efficiency, reduce costs, and improve safety measures. Besides monetary savings, the benefits include increased customer satisfaction, better operational efficiency, and less negative environmental impact through preventing fuel wastage.

2. Business case

2.1 Problem

Airbus currently uses automatic fuel leak detection systems that monitor fuel volume within the tanks. Unfortunately, these systems are inaccurate due to errors in fuel volume measurement in the tanks. This lack of precision makes it difficult to identify leaks caused by issues like:

- Tank Sealant Degradation: Sealant in the fuel tank can degrade due to factors like age, temperature or exposure to harsh chemicals. Sealant degradation can create cracks or gaps, allowing fuel to leak.
- Fuel Tank Structural Damage: External factors such as bird strikes, manufacturing defects, or temperature can cause structural damage to the fuel tanks themselves. This damage can lead to fuel leaks of different magnitudes.

Currently, the problem is that current fuel volume measurements are not sensitive enough to capture small changes in volume caused by these types of leaks, which if left unnoticed can potentially lead to problems with severe safety and financial consequences.

2.2 Objective

The objective for this project is to make use of data analytics to propose an automatic fuel leak detection model, which would allow Airbus to reduce maintenance costs and aircraft availability.

2.3 Current fuel leak detection methods

Currently, there are different methods that can be used for early-detection of fuel leakages. Different procedures are in place based on the different phases of the flight: before take-off, cruise phase, after landing.
The scope of this project is to implement a way to predict fuel leakages based on data and analytics, therefore it is important to focus on the cruise phase of the airplane.

2.3.1 Before the flight

Before take-off, a check is performed to calculate the initial amount of fuel on board:

$$\textit{Initial Fuel On Board (FOB)} + \textit{Fuel Uplifted} = \textit{Fuel On Board (FOB)} \pm \Delta^{1}$$

Initial FOB is the quantity given by the Fuel Quantity Indication (FQI) system. Fuel Uplifted is the amount of fuel added by the refuelers. Δ represents an acceptable tolerance.

2.3.2 During the flight

During the flight, fuel checks are performed mainly to avoid any abnormal fuel consumption. According to current SOPs, three different types of check should be performed every thirty minutes or at each waypoint:

1. Fuel on Board (FOB): Check every difference between FOB and flight plan prediction which may indicate fuel over-burn or external leakages.

2. Flight Management System (FMS) predictions: FMS prediction do not take into consideration any degraded state of the aircraft, making it the ideal baseline for fuel consumption indication. In case of fuel leak, the prediction of FOB at destination and the actual FOB decrease at the same rate drifting away from the initially predicted consumption.
3. Check FOB / Fuel Used: Consists of checking that the sum of FOB and Fuel Used is constant with the Initial FOB. If the sum is unusually smaller than Initial FOB, it may indicate a fuel leak.

$$\textit{Fuel On Board (FOB)} + \textit{Fuel Used} = \textit{Initial Fuel On Board (FOB)} \pm \Delta^{1}$$

2.3.3 After the flight

It is important to perform a fuel check once the aircraft is parked, to check for eventual discrepancies. The post flight fuel check consists of checking that:

$$\textit{Fuel On Board (FOB)} + \textit{Fuel Used} = \textit{Initial Fuel On Board (FOB)} \pm \Delta^{1}$$

2.4 Costs of fuel leakages

Fuel leaks in aircraft can lead to significant financial burdens, including high costs associated with Aircraft on Ground (AOG), maintenance expenses, and reputational damage. Addressing these leaks promptly and efficiently is crucial to minimizing their impact on airline operations and profitability.

2.4.1 Aircraft on Ground costs

AOG (Aircraft on Ground) is an aviation term referring to the situation where an aircraft is out of service or grounded due to technical reasons. An aircraft that is grounded due to technical malfunctions, such as fuel leaks, incurs significant costs. These costs include both direct and indirect expenses that impact the airline's operations and profitability. A breakdown of these costs is as follows:

- Lease Payments: Airlines continue to incur lease payments even when the aircraft is not in service. These fixed costs add financial strain, especially when the aircraft is grounded for extended periods.
- Labour Costs: Fuel leaks often necessitate urgent repairs, leading to overtime pay and emergency call-out fees for maintenance crews.
- Logistical Costs: In some cases, airlines might need to operate additional flights to accommodate displaced passengers, incurring further operational costs.
- Loss of Revenue: While the aircraft is grounded, the airline misses out on potential revenue that would have been generated from ticket sales and ancillary services. Grounded aircraft affect overall fleet utilization, reducing the efficiency of the airline's operations and impacting profitability.
- Insurance and Liability: Frequent AOG incidents can lead to increased insurance premiums as insurers reassess the risk profile of the airline. In the event of a fuel leak leading to accidents or injuries, the airline may face liability claims and legal expenses.

2.4.2 Maintenance costs

Maintenance costs due to fuel leaks include immediate repair expenses and the cost of replacement parts. Sourcing and transporting spare parts can be logistically challenging and expensive. Additionally, the repair process may require highly skilled technicians, resulting in higher labor costs. Fuel leaks often necessitate urgent repairs, leading to overtime pay and emergency call-out fees for maintenance crews.

2.4.3 Disruption costs

Airlines need to compensate passengers for delays, cancellations, and any inconvenience caused, including accommodation and meals. This must be done when an airplane cannot fly due to a fuel leak. Furthermore, there are non-monetary costs such as damage to the airline's brand image and reputation. Passengers may lose trust in the airline, leading to a decline in future ticket sales.

2.5 Estimated impact

Detecting fuel leakages presents a significant opportunity for Airbus to reduce costs and improve operational efficiency. Automatically detecting fuel leaks early will be a game-changer, directly reducing the frequency of Aircraft on Ground (AOG) incidents and delays, while also decreasing overall maintenance costs.

The estimated monthly savings per aircraft are $34,405. This comprises $10,238 from AOG cost reduction and $24,167 from maintenance cost reduction. The AOG savings result from decreasing AOG incidents by 0.5% of flights, while the maintenance savings come from a projected 10% reduction in annual maintenance costs.

Our estimations are based on the following assumptions:

- Each aircraft operates 90 flights per month
- Current AOG rate is 3% of flights[11]
- Average AOG cost per incident is $22,753[3]
- Current annual maintenance cost per aircraft is $2.9 million[2]

While the financial impact of our solution is substantial, the non-monetary benefits are equally significant and far-reaching. Airlines can expect to see improvements across many aspects of their operations, including:

- Reduction of serious incidents: Enhancement of safety measures and lowering the risk of potentially hazardous leakages.
- Operational efficiency: Reduction in unexpected maintenance issues and AOG incidents, allowing for optimized flight schedules and resource allocation. These benefits lead to smoother operations, reduced disruptions, and more efficient use of both aircraft and personnel.
- Customer satisfaction: Customers experience the benefits of a fuel leak detection system through fewer delays and cancellations. This enhanced reliability contributes to a more positive travel experience, potentially leading to increased customer loyalty and a better brand reputation.
- Environmental impact: Prevention of fuel wastage, saving costs and aligning with the growing environmental concerns in the aviation industry, supporting efforts to reduce Airbus's carbon footprint.

Boeing has been facing significant reputation issues in 2024, with a series of incidents that have raised concerns about the company's commitment to safety and quality control. For instance, in January, an Alaska Airlines 737 MAX lost a door plug mid-flight, leading to a terrifying experience for passengers and subsequent scrutiny from the FAA. This was followed by reports of loose bolts and hardware on other 737 MAX planes, and a United Airlines 737 MAX experiencing stuck rudder pedals during landing procedures.

The cumulative effect of these incidents has severely shaken public confidence in Boeing's aircraft. The FAA has mandated that Boeing develop a comprehensive plan to address these systemic quality-control issues, further emphasizing the gravity of the situation. Moreover, the company has failed a significant number of audits, revealing deep-rooted problems in its manufacturing processes.

The reputation crisis at Boeing provides one critical lesson to our project about the need for meticulous and clear safety procedures, especially in detecting fuel leakages within aircraft. Our project is going to address one of the critical safety issues: developing a model for fuel leak detection. Making this detection system reliable and accurate will help us prevent incidents similar to Boeing's recent troubles and maintain high safety standards in aviation.

3. Data Exploration

3.1 Data explanation

We were provided historical datasets containing time-series data coming from the sensors on the planes for every flight. One dataset was for a flight test aircraft (MSN02) and included 111 features, while the other seven datasets were for seven in-commission aircrafts and included 17 features. Within all of these datasets together, there were 23,990,158 rows of time series data.

For the seven datasets with 17 features, the features focused on two types of data:

1. Aircraft and flight data:
    a. UTC date/time
    b. Aircraft name
    c. Flight number
    d. Flight phase (from pre-flight to descent)
    e. Altitude
2. Fuel and engine system data
    a. Fuel used, by engine
    b. Total fuel on board (volume in Kg)
    c. Fuel quantity per collector cell and surge tank (volume in Kg)

In addition to the above, MSN02 also included features on many additional topics. It must be noted that we did not have a data dictionary for many features in the MSN02 dataset, and so had to deduce what many of them meant from the feature names. These features include:

a. Pitch and roll
b. Engine status (running or not)
c. Fuel flow to each engine
d. Pump status (on/off, normal/abnormal, immersed/not immersed)
e. Fuel transfer mode

Certain features have a very high rate of nulls. Features on the amount of fuel used have an overall 91% null rate across the entire dataset. Meanwhile, fuel on board features have an overall 4% rate across the entire dataset.

3.2 Data cleaning

The first step of any analysis is to clean up the data. We began here, by investigating what portion to use in our analysis.

3.2.1 Handling nulls

As previously mentioned, certain features have a very high rate of nulls. Features on the amount of fuel used have an especially high null rate, at a 91% rate of nulls overall. While we considered methods of imputing these values, we decided that imputing 91% of the values of a feature that feeds into our model was too much guesswork. Therefore, we decided to simply remove all data entries that had null values. After dropping these null rows, we were left with 904,152 rows of data.

3.2.2 Flight phase 8

The data includes entries from the entire lifecycle of a flight. The flight phases include pre-flight, engine run, various stages of take-off and climbing, cruise, descent, approach, landing, and post-flight. One issue we ran into early on is that during many of these phases, an aircraft is subject to a lot of turbulence or other inconsistent conditions that could cause incorrect or volatile fuel measurements. After closely investigating these different phases, we decided that phase 8, cruise, made the most sense on which to focus. During many of the other phases such as take-off and descent, the aircraft's pitch is by definition much steeper; and during taxi phases such as pre-flight and postflight, the aircraft is accelerating, turning, and decelerating all the time. All of these activities create the possibility for inconsistent fuel measurements. During cruise, however, an aircraft's pitch, roll, and acceleration will be much more stable, allowing much easier identification of inconsistencies that could indicate a fuel leakage. We therefore filtered our data to only utilize flight phase 8 in our model. After filtering, we were left with 376,553 rows of data and 479 flights.

3.2.3 Outlier handling

During our analysis, we identified numerous outliers in both the fuel used and the fuel on board values. These outliers included implausible entries, such as the fuel on board dropping to zero for several records before returning to realistic levels. We opted not to manually handle these anomalies to ensure our model is exposed to the full spectrum of data variations, including extreme outliers. By retaining these significant anomalies, we leverage the model's capability to detect outliers naturally, as it is designed to identify and react to these abrupt deviations in the dataset. This approach allows our model to learn and adapt to the presence of such anomalies, enhancing its robustness and accuracy in identifying genuine fuel leakages.

3.3 Exploratory Data Analysis

After cleaning the data, our next phase was data exploration, where we tested hypotheses and investigated the data to understand its contours better. Some avenues were dead ends and others provided understanding about the data we could apply to future analyses. While we cannot detail every exploratory analysis we conducted, we discuss several key ones next.

3.3.1 Flight test aircraft

As previously mentioned, we were provided with one dataset on a flight test aircraft, MSN02, which had far more features than the other datasets. One of our first avenues for data exploration was to try to understand the contents of this data. While we of course could not directly apply learnings about particular features that were present in this dataset but not in the other datasets, we wondered whether we could learn trends or patterns from these extra features that would inform our understanding of the other datasets.

First, we focused on minimizing the number of features with which we were working. We began by combining those that could be combined; for example, we created the feature Num_Engines_Running from the features `ENGINE_RUNNING_1`, `ENGINE_RUNNING_2`, `ENGINE_RUNNING_3`, and `ENGINE_RUNNING_4`. Certain True/False features had identical values by row to other related features, so we could combine the information into just one feature; for example, `STATE_PMP_MAIN_FT1_IMMERSED`, `STATE_PMP_STBY_FT1_IMMERSED`, `EF1_Density`, and seven other features had identical values within each row and so were combined into one feature entitled `STATE_MAIN_STBY_IMMERSED`. We also dropped all features that only had one possible value in the dataset outside of nulls, as we knew that the lack of differentiation would make this feature useless; these features included `STATUS_FUEL_QTY_UNUSABLE_RST`, `STATUS_FUEL_QTY_UNUSABLE_LST`, `STATUS_OVERFLOW_RST`, and `STATUS_OVERFLOW_LST`. We know that the leak indicators in this dataset, `STATUS_FUEL_LEAK_DETECTED_VALID` and `LEAK_DETECTION_LEAK_FLOW`, are largely untrustworthy, so while we tested these a bit, we largely ignored them.

After dropping and combining the number of features down from 111 to 68, we next conducted some analyses. We created correlation matrices to identify additional features we could drop due to high correlation with other features.

As can be seen in Figure A1 (in the Appendix), it was easy to note features that are highly correlated with other features. Utilizing this method, we removed additional features from the MSN02 dataset.

After significant investigation, it was decided that there were few learnings to take away from the MSN02 dataset that could be applied to the other datasets. Additional analyses on MSN02 data will be discussed later in conjunction with more advanced analysis techniques.

3.3.2 Fuel imbalance

Upon further research, we learned that commonly, pilots take note of any imbalances between the fuel quantities in the left and right sides of the wing to indicate a potential fuel leakage.[7] We decided to investigate similar imbalances to determine whether it could assist us in identifying fuel leaks.

We began by creating a new variable `diff_FOB` which was calculated using the following formula:

```
diff_FOB = |(VALUE_FUEL_QTY_FT1 + VALUE_FUEL_QTY_FT2 + VALUE_FUEL_QTY_LXT) -
        (VALUE_FUEL_QTY_FT3 + VALUE_FUEL_QTY_FT4 + VALUE_FUEL_QTY_RXT)|
```

This formula takes the absolute value of the difference between the sum of the fuel on board in every tank on the left side of the wing and the sum of the fuel on board in every tank on the right side of the wing. After calculating this value for each time series entry, we plotted the distribution in a histogram.

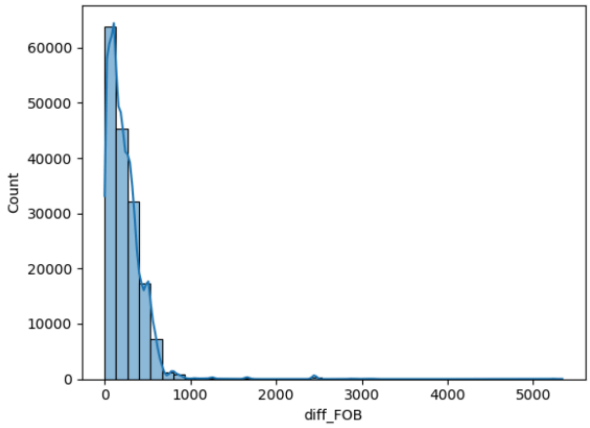

Figure 1. A histogram of the difference in fuel on board between left and right sides of the wing, in kilograms.

As shown in Figure 1, the vast majority of differences in fuel on board are low-level. In fact, the mean difference is 235 kilograms. However, there are quite a lot of instances of larger difference, with the majority below 1,000 kilograms and a long tail of instances after that.

3.3.3 Gradient of flight

As discussed previously, we know that a change in pitch angle of the aircraft is likely to affect fuel measurements. However, we logiced that the cruise flight phase could occasionally be turbulent as well, caused by mountains or storms.[6] But pitch angle is only available in MSN02 data - so how do we come up with a similar proxy for the other datasets?

These datasets consist of time series data, so we have data for every individual second of a flight. We also have altitude data. By calculating the difference in altitude between consecutive seconds of a flight, we can approximate turbulence through understanding how much an aircraft dropped or rose in a short period of time.

We then compared altitude change against our previously-calculated fuel imbalance metrics. We were curious to see if increased turbulence could cause some sensor mishaps that would cause sensors on one side of the plane to read higher than on the other, such as if fuel sloshed in one direction.

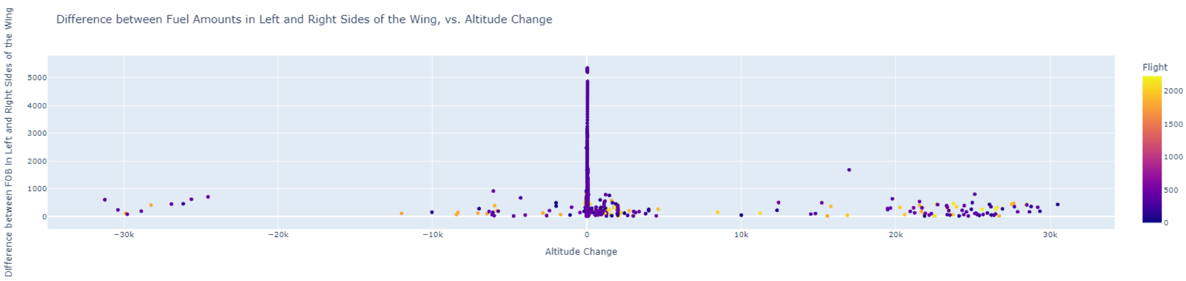

Figure 2. A scatterplot of Altitude Change as compared to Difference between FOB in Left and Right Sides of the Wing.

As can be seen in Figure 2, there are no patterns that indicate that altitude change is correlated to a fuel imbalance. This line of investigation was therefore dropped.

3.3.4 Principal Component Analysis (PCA)

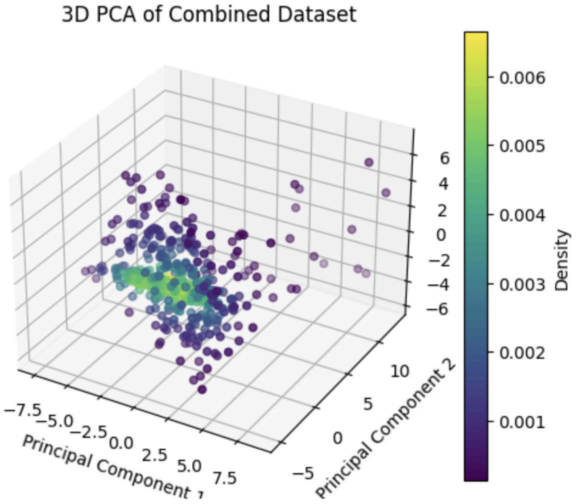

Figure 3. 3D PCA of the dataset.

We applied Principal Component Analysis to identify key features and reduce dimensionality. While we knew that `VALUE_FOB` and `total_fuel_used` were important, PCA helped us explore additional significant variables.

The 3D scatter plot visualizes PCA results, where each point represents a data instance, with color indicating data density. This helps identify clusters and trends, revealing underlying structures or anomalies.

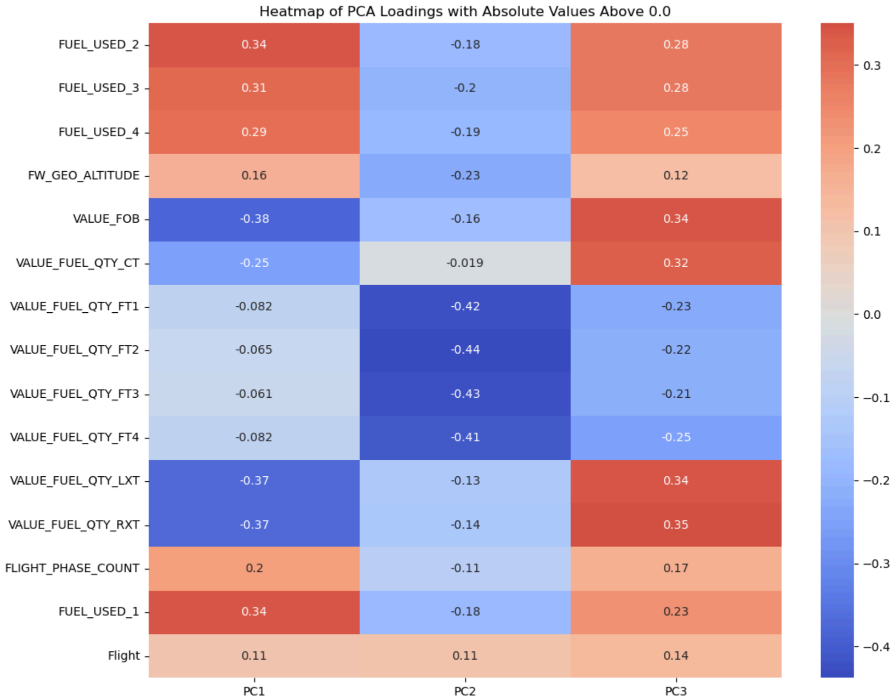

Figure 4. Heatmap of the PCA Factor Loadings.

From the heatmap of PCA loadings, we observed that the first principal component (PC1) is primarily driven by `FUEL_USED_1`, `FUEL_USED_2`, `FUEL_USED_3`, `FUEL_USED_4`, and `VALUE_FOB`, highlighting variations in fuel usage and onboard quantities. The second principal component (PC2) is influenced by `VALUE_FUEL_QTY_FT1`, `VALUE_FUEL_QTY_FT2`, `VALUE_FUEL_QTY_FT3`, and `VALUE_FUEL_QTY_FT4`, which focus on specific tank fuel quantities. The third principal component (PC3) is affected by `VALUE_FOB`, `VALUE_FUEL_QTY_CT`, and `VALUE_FUEL_QTY_LXT`, showing another aspect of fuel distribution.

Despite these insights, we decided not to use PCA further as we did not find it beneficial for our specific needs.

3.4 Feature engineering

In the feature engineering phase of the project, we derived new features to enhance how informative the dataset is and to improve the model's predictive power. We list below the new features, their formulas, and why they were measured.

New Feature	Formula	Measurement
`total_fuel_used`	`FUEL_USED_1 + FUEL_USED_2 + FUEL_USED_3 + FUEL_USED_4`	A comprehensive view of the total fuel consumption.
`initial_FOB`	`df.groupby('Flight')['VALUE_FOB'].transform('first')`	Initial FOB for each flight.
`fuel_used+FOB`	`'total_fuel_used' + 'VALUE_FOB'`	Sum of total fuel used and the current FOB.
`fuel_used_per_minute_total_fuel_used_diff`	`'initial_FOB' - 'fuel_used+FOB'`	Minute-by-minute difference in total fuel used.
`diff_FOB`	`'VALUE_FOB'.diff()`	Minute-by-minute difference in the Fuel on Board (FOB).
`diff_initial_FOB_fuel_used+FOB`	`'initial_FOB' - 'fuel_used+FOB'`	Compares the initial FOB with the sum of the total fuel used and current FOB to identify discrepancies.
`predicted_FOB`	`'initial_FOB' - 'total_fuel_used'`	Estimate of the expected FOB.
`Real_FOB-predicted_FOB`	`'VALUE_FOB' - 'predicted_FOB'`	Helps identify potential anomalies indicative of fuel leakage by comparing FOB with predicted FOB.
`flight_entry`	`df.groupby('Flight').cumcount()`	Marks the sequence of data entries for each flight, aiding in time-series analysis to identify leakages.

Table 1. Feature engineering conducted on the datasets.

4. Data manipulation

After exploring the data, we knew we needed to decide next how to manipulate the data so we could then train a predictive model on it. Because we had no labeled data, we decided to start by implementing synthetic leakages in the dataset.

4.1 Implementation of synthetic leakages

By simulating leaks, we have complete control over the characteristics of the anomalies, making it easier to analyze the model's performance and allowing us to test various leakage scenarios.

4.1.1 Objective

The primary objective of implementing fake fuel leakages was to create a controlled environment to train and validate our model. Given the rare occurrence of actual fuel leaks, generating synthetic leak events allowed us to ensure that the model had enough instances to learn from.

4.1.2 Leakage simulation function

We designed a function to simulate a leakage event starting mid-flight. The simulation function aimed to mimic real-world leak scenarios by introducing gradual fuel loss over time. The steps involved include:

1. Leak Duration Calculation: We determined the number of time steps from the start point of the leak to the end of the flight. This calculation ensured that the leak simulation spanned a realistic duration, allowing the model to detect gradual changes in fuel metrics.
2. Fuel Loss Calculation: We generated a sequence of incremental fuel losses over the leak duration based on a predefined leak rate. The leak rate, set at 0.5 kg/s per time step, was chosen to create a noticeable but realistic fuel loss. This incremental approach ensured that the simulated leaks reflected real-world scenarios where fuel loss occurs gradually.
3. New Column: We subtracted calculated fuel loss from Fuel on Board and stored the result in a new column `FAKE_LEAKAGE`. This column represented synthetic anomalies, allowing us to train the model with clear examples of fuel leakage events.

4.1.3 Random flight selection

To ensure variability and robustness in our model, we randomly selected 100 flights from the dataset to apply the synthetic leakages. This random selection process was crucial to mimic real-world conditions where leaks could occur in any flight under various scenarios. We used a random seed (42) for reproducibility, ensuring that the selected flights could be consistently replicated in future analyses. The diversity of selected flights allowed the model to encounter a wide range of operating conditions, improving its ability to generalize and detect anomalies across different flight profiles.

4.1.4 Application of simulation

For each of the selected flights, we determined a random starting point for the leak, typically mid-flight. This approach ensured that the simulated leaks occurred during stable flight conditions, making it easier to isolate and detect the anomalies. We conducted the following:

1. Extract Flight Data: We extracted the time series data for each selected flight, focusing on the relevant fuel metrics.
2. Leakage Simulation: We applied the leakage simulation function to generate the FAKE_LEAKAGE column for these flights. This process involved calculating the incremental fuel loss and adjusting the Fuel on Board accordingly.
3. Update the Dataset: We updated the main dataset with the simulated leakage data, ensuring that the FAKE_LEAKAGE column accurately reflected the synthetic anomalies for each selected flight.
4. Export the Data: The updated dataset was exported to preserve the modifications and to use the enhanced dataset for training and validation purposes.

This process ensured that each selected flight had a unique leakage event, varying in start time and progression, thereby providing a comprehensive set of anomalies for the model to learn from.

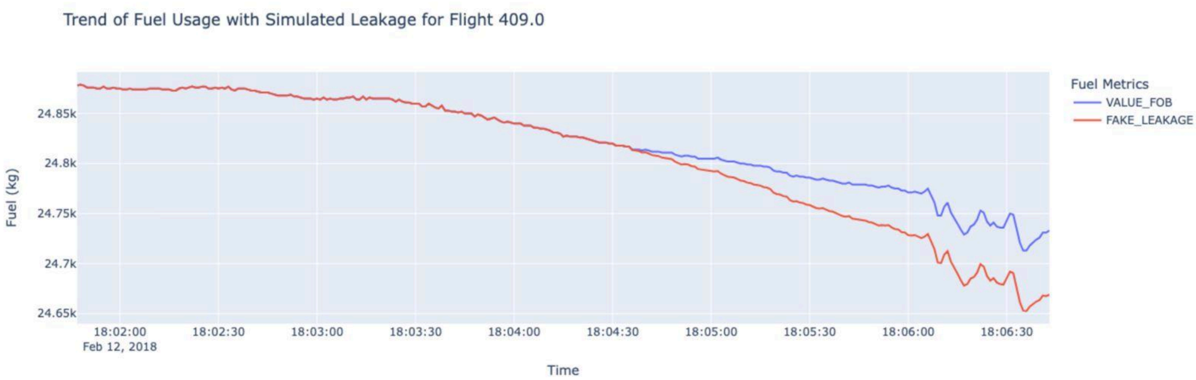

Figure 5. Fuel on Board plus synthetic leakage implemented on Flight 409.

5. Solutions

The next phase was to decide which model to use. As we considered our options, we adopted a time-based splitting strategy for our dataset, which consists of time-series data with essential temporal dependencies. The split was performed using a designated date, 1 February, 2018, with the goal of achieving an approximately 80/20 distribution between the training and testing datasets. This approach was chosen to maintain the chronological order of flights, thereby preserving the integrity of temporal patterns and potential dependencies within the data. Both the training and testing datasets include instances of leakages, ensuring that our model is robustly evaluated across varied scenarios and conditions.

5.1 Solution options

Throughout our model creation process, we considered various approaches to predicting these fuel leakages. We will next lay out several of the most interesting options we considered. (See confusion matrices for these model options in the Appendix.)

5.1.1 Autoencoder

One approach we took was using autoencoders. We considered autoencoders for anomaly detection due to their ability to identify subtle patterns and anomalies in complex, noisy datasets like our airplane fuel usage data. Autoencoders compress the data into a lower-dimensional space and then reconstruct it, learning normal patterns during training. This method is robust to noise, making autoencoders ideal for detecting fuel leakages amidst the inherent noise of aircraft operational data. (See Figure A2 in Appendix.)

5.1.2 Isolation Forest

We chose Isolation Forests as another anomaly detection option due to their effectiveness in identifying anomalies in complex, high-dimensional datasets. Isolation Forests operate by randomly partitioning the data and isolating observations. During this process, normal data points require more partitions to isolate, while anomalies, such as potential fuel leaks, are isolated with fewer partitions because they deviate significantly from the majority of the data. This method is particularly robust to the presence of noise and can efficiently handle large datasets, making Isolation Forests well-suited for detecting fuel leakages amidst the inherent noise in airplane operational data. (See Figure A3 in Appendix.)

5.1.3 XGBoost

Another approach we explored was XGBoost, a powerful and flexible gradient boosting framework. XGBoost is known for its high performance and efficiency in predictive modeling tasks, particularly with structured data. We considered XGBoost for predicting fuel leakages due to its ability to handle complex patterns and interactions within the dataset. By iteratively adding trees and optimizing the model, XGBoost can effectively capture both linear and non-linear relationships, which is crucial for identifying the subtle indicators of fuel leakages. Moreover, XGBoost includes regularization parameters to prevent overfitting, enhancing its robustness to noise and ensuring reliable performance on our aircraft operational data. This makes XGBoost an excellent candidate for accurately detecting fuel anomalies amidst the inherent noise in the dataset. (See Figure A4 in Appendix.)

5.2 Our solution

After much exploration, our work with Autoencoders had better metrics than the Isolation Forest option, so we decided to go with the Autoencoder option. Initially, we developed a basic autoencoder as a baseline. The architecture included an input layer, an encoding layer with twelve neurons and a TanH activation function, and a decoding layer with a sigmoid activation function. We used the Adam optimizer and Mean Squared Error (MSE) loss function. The autoencoder was trained on standardized training data for 50 epochs with a batch size of 10, including validation on test data. We set the threshold for reconstruction error at the 75th percentile to classify anomalies, converting reconstruction errors into binary anomaly labels (1 for anomaly, 0 for normal) based on this threshold. We then adjusted the threshold to minimize false positives, as this is the most costly and undesirable outcome in our situation.

This set of parameters did not provide as good a performance as we were aiming for, so we created a new autoencoder to optimize our results and detect more of the leakages. We designed a more complex autoencoder architecture aimed at enhancing its ability to capture intricate patterns and anomalies within the data. The new autoencoder consists of an encoder with layers that start with 80 neurons and gradually reduce to 50, each employing the Rectified Linear Unit (ReLU) activation function to enhance feature extraction. The decoder mirrors this structure, progressively expanding back to 80 neurons before projecting the reconstructed output with a final layer employing the sigmoid activation function. We chose this architecture to ensure robust encoding and decoding of the temporal patterns present in our time-series data. The autoencoder was trained over 300 epochs using a batch size of 100, with training and validation performed on separate datasets split chronologically, thus preserving the temporal integrity crucial for accurately identifying anomalies. The confusion matrix from this approach can be seen below.

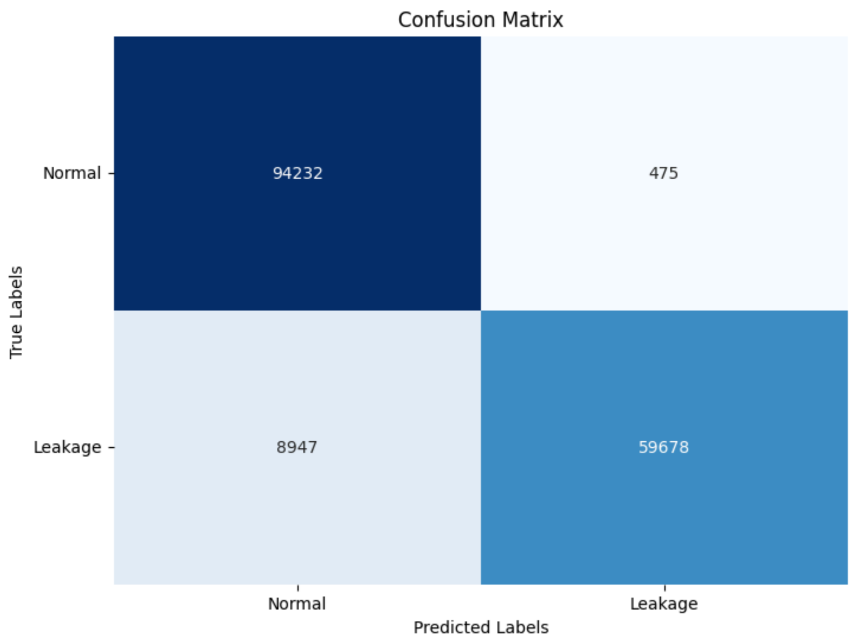

Figure 6. The confusion matrix resulting from the final Autoencoder approach.

We managed here to drastically improve our leakage detection, with metrics of:

- Accuracy: 94.24%
- Recall: 86.95%
- Precision: 99.21%
- F1 Score: 92.60%

6. Business implementation

The model we created will be implemented on aircrafts by the creation of a user-friendly dashboard. The tool is specifically designed for both aircraft crew and ground staff. For aircraft crew, the dashboard will be implemented into the aircraft system, with data points fed directly to the dashboard once the plane completes the flight.

In the initial phase, the dashboard will be available to a small group of pilots operating on high-risk routes, such as long-haul flights or flights which are known to operate routes with high turbulence, and for older aircrafts or aircrafts which had leaks in the past. Additionally, a small group of aircraft mechanics, maintenance staff, and inspectors will also be included in the initial deployment phase, to aid in continuous maintenance of the aircraft.

The second phase will consist of collecting data and feedback from the initial users and implementing changes to the dashboard. Finally, the dashboard will be made available to the whole fleet and to the ground staff. In order to do so, training will be offered to all users.

6.1 Dashboard

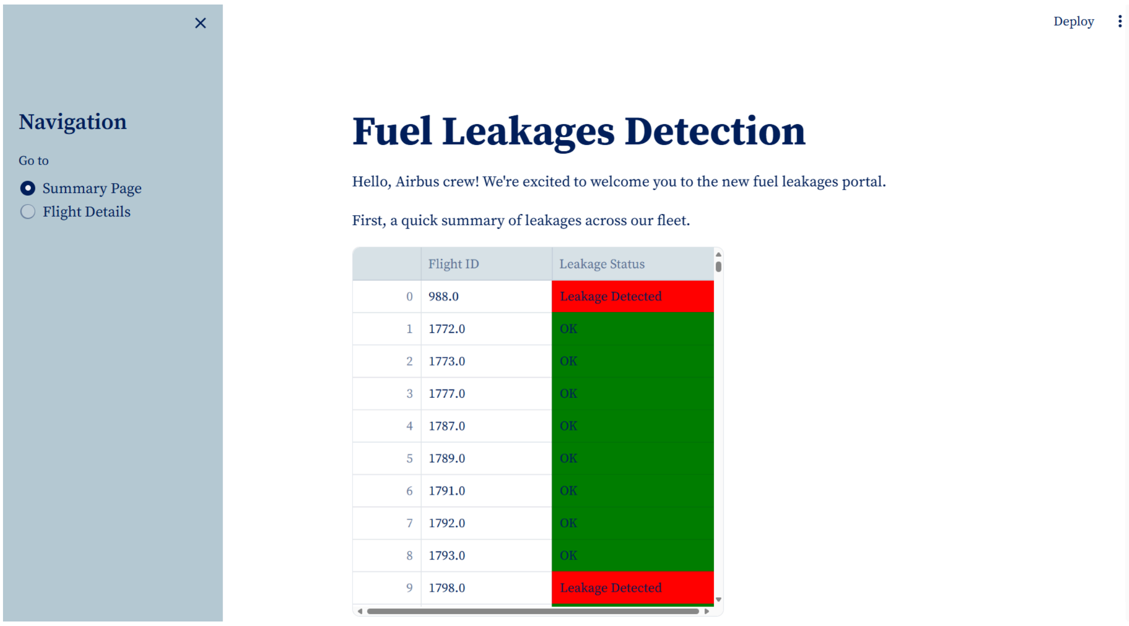

Figure 7. Screenshot from the first page of the dashboard.

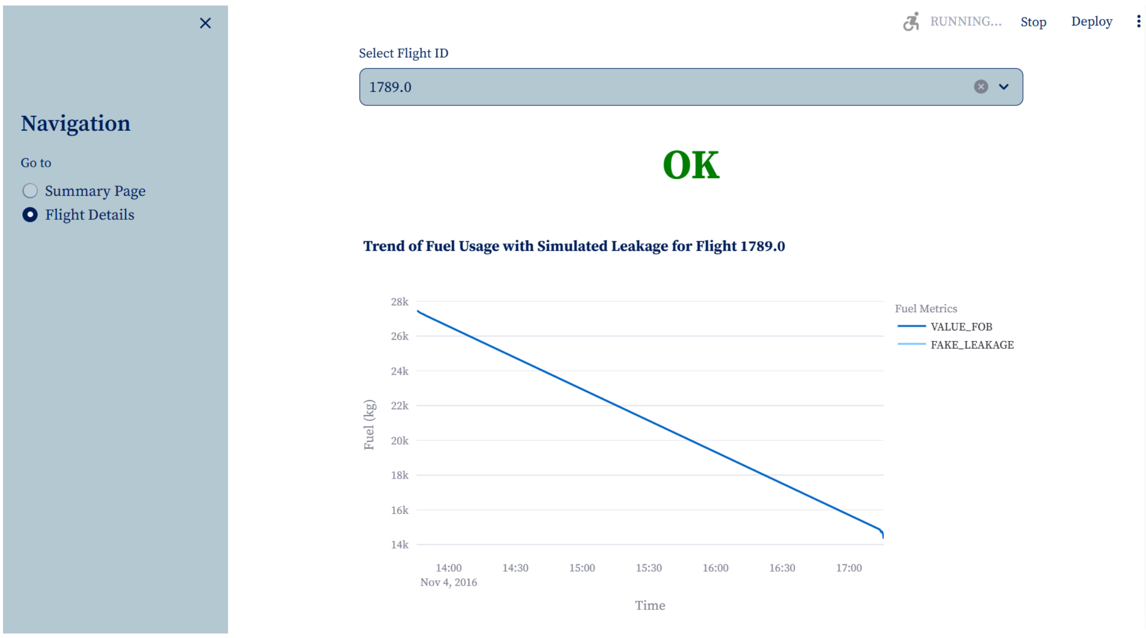

Figure 8. Screenshot from the second page of the dashboard.

Our current dashboard implementation is designed to detect fuel leakages in historical flight data, providing a retrospective analysis based on the recorded flight data. This allows for post-flight evaluations, identifying any instances of fuel leakage after the flight has concluded. The dashboard leverages our predictive models to analyze the historical data, offering insights and detailed reports on fuel usage anomalies.

Looking ahead, our vision for the dashboard is to enhance its capabilities to perform real-time data analysis. The ultimate goal is for the model to process data in real time during flights, enabling immediate detection of fuel leakages. Upon identifying a leakage, the system will promptly notify the pilot in the cockpit, ensuring timely intervention. While the current implementation is limited to historical data analysis, our ongoing development efforts are focused on achieving real-time monitoring and alerting to significantly improve flight safety and operational efficiency.

7. Conclusion

This report presents a comprehensive approach to predicting and detecting fuel leakages in Airbus aircraft, aimed at enhancing safety, reducing maintenance costs, and improving operational efficiency. The current methods of fuel leak detection, which primarily focus on monitoring the volume of fuel in the tanks on each side of the aircraft, are often inaccurate and fail to detect small but critical leaks.

Our extensive data exploration process highlighted the importance of focusing on stable flight stages, particularly the cruise phase, for accurate fuel leakage detection. This allowed for the injection of synthetic leakages to create a robust dataset for model training and validation. Eventually, the Autoencoder model was chosen for its superior performance in anomaly detection, achieving high metrics. The model is integrated into a user-friendly dashboard, which we plan to initially deploy to pilots on high-risk routes and older aircraft, followed by a fleet-wide rollout. This implementation is expected to enhance operational efficiency, reduce costs, and improve safety measures.

A first future improvement is to shift from the current post-flight data analysis toward real-time monitoring during flights. Real-time data processing, with instant alerts to the flight crew, would allow for quick responses against any potential leaks. Plus, additional or advanced sensors could be added to aircraft to perform leak detection more accurately. This step would require the investigation of new technologies in fuel sensors that give better resolution and reliability in different flight conditions. Lastly, continuous learning models that adapt based on new data could also enhance the system’s robustness and accuracy over time.

Much more extensive field testing in a variety of flight conditions and in many different aircraft types will provide a much better validation of the model. A feedback loop in which pilots and maintenance crews provide feedback on the system is critically important. Additionally, an environmental impact analysis will give additional support for investment in this technology by quantifying the system's benefits in fuel savings and reduced carbon emissions. Focusing on these areas of future improvement puts the system well on track to being a sophisticated, reliable, and integral part of aircraft operation and maintenance at Airbus.

8. References

1. Airbus (2024) 'Fuel monitoring on A320 Family aircraft,' *Safety First*, 1 July. https://safetyfirst.airbus.com/fuel-monitoring-on-a320-family-aircraft/.
2. *Airline Maintenance Cost Executive Commentary: FY2022 Data* (2024). International Air Transport Association. https://www.iata.org/contentassets/8437020db31a4717b70677d9b06b1a45/fy2022-mcx-report_public.pdf.
3. Badkook, B. and Badkook, E. (2016). Study to Determine the Aircraft on Ground (AOG) Cost of the Boeing 777 Fleet at X Airline. *American Scientific Research Journal for Engineering*, [online] 25(1), pp.51–71. Available at: https://asrjetsjournal.org/index.php/American_Scientific_Journal/article/download/2110/881/6017.
4. Beresnevičius, R. (2024) *How Much Do Aircraft Groundings Cost Airlines & Manufacturers?* https://simpleflying.com/aircraft-grounding-cost-to-airlines-and-manufacturers/.
5. Bnomozbaev (2022a) 'AOG (Aircraft on Ground),' *Delta Global*, 19 January. https://www.deltaglobalsolutions.com/blog/aircraft-on-ground.
6. Donahue, M.Z. (2023) 'What causes turbulence, and what can you do if it happens to you?,' *Travel*, 3 March. https://www.nationalgeographic.com/travel/article/what-is-turbulence-explained.
7. *Fuel Leak and Confirmation Bias (OGHFA SE)* (2022). https://skybrary.aero/articles/fuel-leak-and-confirmation-bias-oghfa-se.
8. *How much does it cost to have an aircraft on ground (AOG)?* (no date). https://www.eways-aviation.com/blog/how-much-does-it-cost-to-have-an-aircraft-on-ground-aog.
9. Iagcargo (2018) *Aircraft on Ground: How a technical fault can cost millions*. https://iagcargomagazine.com/2018/07/03/aircraft-on-ground-how-a-technical-fault-can-cost-millions/.
10. Marani, F. (2008) 'Safety First.' https://safetyfirst.airbus.com/app/themes/mh_newsdesk/documents/archives/low-fuel-situations-awareness.pdf.
11. Layden, Z. (2020). *A SIMPLE GUIDE TO MITIGATE AGAINST AOG LOSSES*. [online] Falcon MGA Services, pp.1–3. Available at: https://falconmga.com/pdf/RE003-Whitepaper_AW-10JUL20_screens.pdf [Accessed 16 Jun. 2024].

9. Appendix

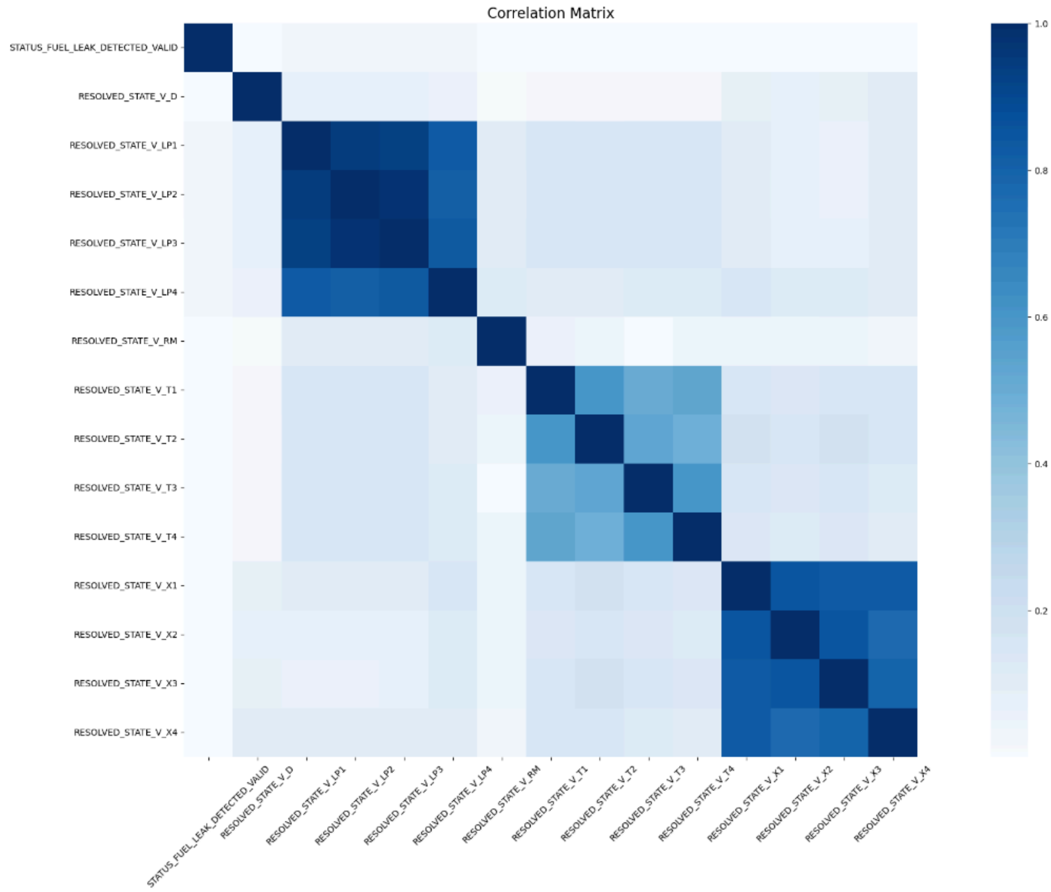

Figure A1. Correlation matrix for 14 features in the MSN02 dataset.

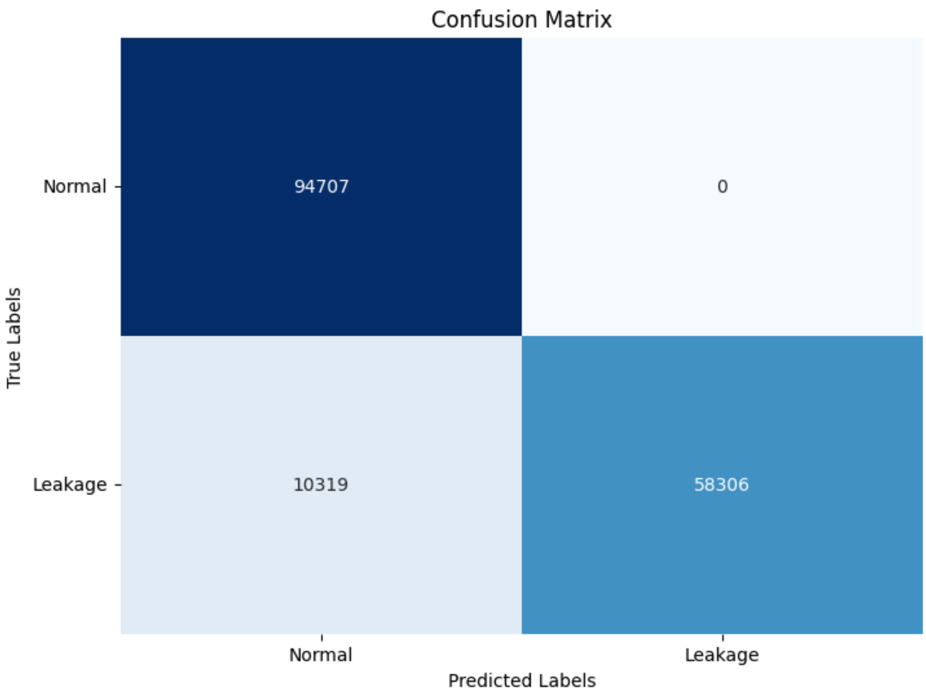

Figure A2. The confusion matrix resulting from an Isolation Forest approach.

From the confusion matrix in Figure A2, the following metrics were calculated:

- Accuracy: 93.68%
- Recall: 84.96%
- Precision: 100%
- F1 Score: 91.87%

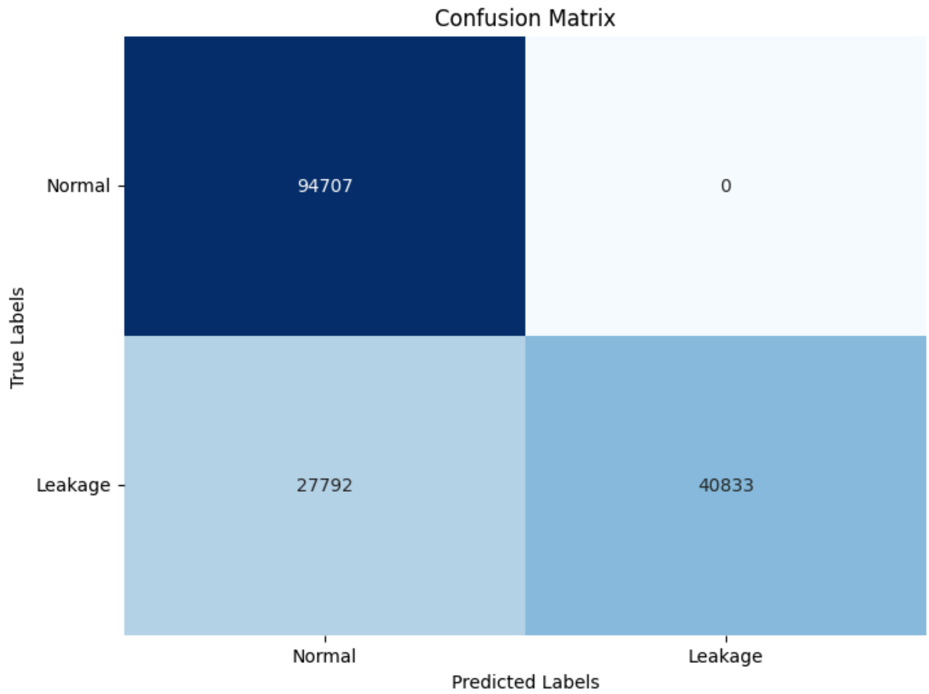

Figure A3. The confusion matrix resulting from the first Autoencoder approach.

From the confusion matrix in Figure A3, the following metrics were calculated:

- Accuracy: 85%
- Recall: 59.50%
- Precision: 100%
- F1 Score: 74.61

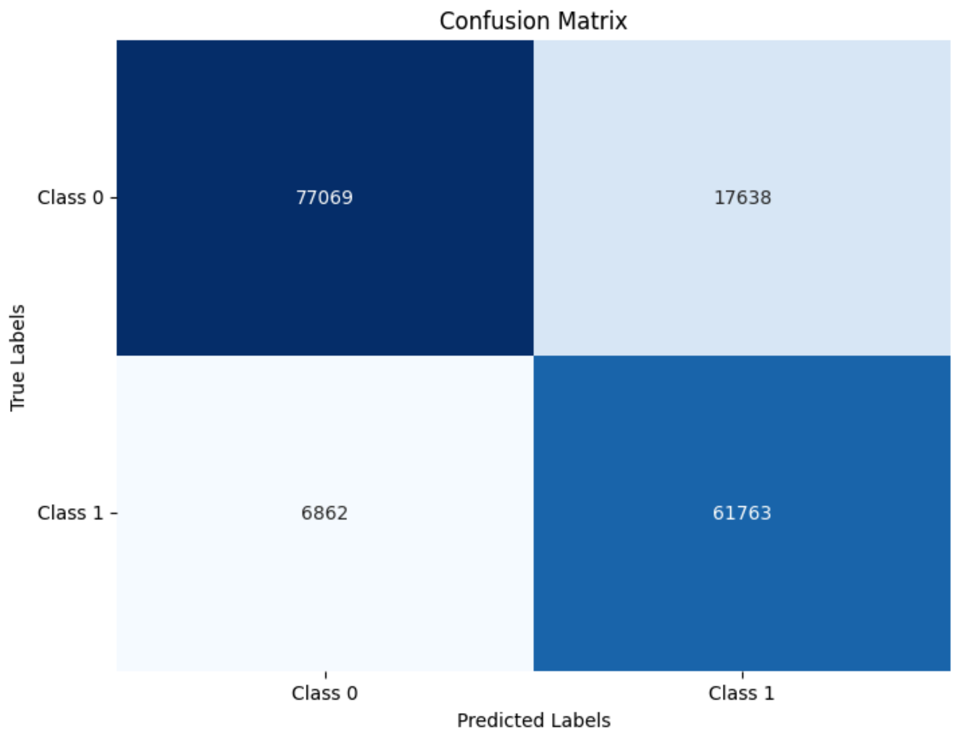

Figure A4. The confusion matrix resulting from the XGBoost approach.

From the confusion matrix in Figure A4, the following metrics were calculated:

- Accuracy: 85%
- Precision: 77.77%
- Recall: 90.00%
- F1 Score: 83.33%