Radar Health Monitoring Using Anomaly Detection

Jean-Marc Divanon, Thomas Lavigne, Theo Cornu, and Teck Yoong Chai

Thales Research & Technology, Singapore

jean-marc.divanon@thalesgroup.com
thomas.lavigne@student-cs.fr
theo.cornu@thalesgroup.com
teckyoong.chai@thalesgroup.com

ABSTRACT

Predictive maintenance has emerged as a crucial strategy in complex systems management, leveraging machine learning and data-driven health monitoring to anticipate failures and optimize operational uptime. While significant progress has been made in developing general-purpose models for anomaly detection and condition-based maintenance, their effectiveness often diminishes when applied to highly specialized systems such as radar platforms. These systems exhibit unique operational behaviours and failure modes, necessitating tailored monitoring solutions. This paper presents a methodology for anomaly detection tailored to radar systems, addressing the inherent challenge of limited labeled data and the ambiguity surrounding the definition of anomalies. We employ a reconstruction-based approach using autoencoders in conjunction with Mahalanobis distance to generate anomaly scores, enabling the detection of subtle deviations from normal system behavior without requiring explicit failure labels. The proposed approach has been applied to real sensor data collected from multiple radar units, specifically from sensors located on the antenna mast. For confidentiality, the data has been anonymized. Experimental results demonstrate that the method effectively highlights outliers and identifies the contributing features responsible for anomalies. Furthermore, the model reveals interpretable abnormal patterns and provides early indications that condition-based monitoring can be a viable strategy for identifying potential issues in radar operations.

1. INTRODUCTION

Condition-based maintenance (CBM) and predictive maintenance strategies are attracting growing interest in various industrial sectors, driven by the increasing availability of sensor data and advances in machine learning (Leukel, Gonzalez, and Riekert, 2021). These approaches enable health monitoring systems to move from reactive to proactive paradigms, identifying signs of degradation or failure before they lead to critical issues. In particular, data-driven models now offer promising capabilities for detecting anomalies and tracking equipment health in real-time.

However, the performance of such models is highly dependent on the quality and completeness of the available data. In practical settings, and especially in complex systems like radar platforms, sensor data can be partial, noisy, or missing altogether — whether due to sensor failures, intermittent system operation, or logging constraints. This makes it difficult to build purely statistical models or to assign simple health status labels with confidence.

Given this challenge, our goal is to develop a robust anomaly detection framework tailored to the specificities of radar systems. Rather than addressing the system as a whole, we focus our work on a particular subsystem: the antenna mast, a critical mechanical component monitored by several sensors.

Our proposed methodology follows a two-stage approach. The first stage involves a *single-feature analysis*, where we analyze each sensor variable individually to understand its behavior over time, detect basic anomalies, and prepare the ground for a more complex analysis. The second stage consists of a *multi-feature analysis*, integrating signals from multiple sensors into a unified detection model, allowing for more refined and context-aware anomaly detection.

Both stages require careful preprocessing to address the imperfect nature of the dataset, including handling of missing data, normalization, and segmentation of sensor logs. This paper is organized as follows: we first describe the radar system, the collected data, and the machine learning architecture used. We then present the single-feature analysis methodology and results, followed by the development and evaluation of our complete multi-feature anomaly detection framework.

2. SYSTEM, DATA & MODEL DESCRIPTION

In this section, we describe the radar systems and the specific subsystem we analyse: the antenna mast. Sensor data is collected from this subsystem and includes both numerical features (such as sensor measurements) and binary signals (status indicators, position states, etc.).

We use data from four radars, anonymized as Alpha, Bravo, Charlie and Delta. These units provide enough data for training and testing our anomaly detection models. In addition to sensor data, we also include maintenance records, which indicate whether an intervention has taken place. However, these logs are limited and do not offer detailed labeling of the system's health.

We will also study the structure of the dataset, the types of features used, and the model architecture designed for anomaly detection.

2.1. System Description

The antenna mast is the mechanical structure that connects the equipped platform to the fixed part of the drive mechanism on which the antenna is mounted. Its role is to raise and lower the antenna between multiple predefined configurations, depending on the operational needs.

The mast can be in one of the following positions:

- **Position 0**: The antenna is fully folded down. It is in a non-operational state, fully retracted for transport or storage. Both the drive actuator and the telescopic actuators are retracted.
- **Position 1**: The antenna is raised just above the shelter roof, allowing access for maintenance or operation. The drive actuator is deployed, while the telescopic actuators remain retracted.
- **Position 2**: The antenna is fully deployed, positioned approximately 8 meters above ground level. In this configuration, the telescopic actuators are deployed, and the drive actuator is retracted.

Based on the mast's mechanical architecture, we collect the following data from the system at a sampling rate of every 10 seconds:

- Pressure values from two telescopic actuators (**actuator_1** and **actuator_2**),
- Pressure from the drive actuator (**actuator_3**),
- Oil temperature from the hydraulic tank,
- Binary indicators reflecting the current position of the antenna mast (positions 0, 1, or 2).

These features form the basis of our analysis in the next sections.

2.2. Maintenance Data

For each radar unit, we have access to three types of maintenance records:

- **Defect Record Sheets (DRS)** – These report faults that were serious enough to require sending the radar system for repair.
- **Maintenance Check Flights (MCF)** – These list minor issues that were fixed directly on-site and did not require the system to be removed from operation.
- **Bite Alerts:** A log of alerts triggered by events, which can range from routine occurrences (such as completing the deployment of the mast antenna) to more significant failures.

Each datasets include useful information such as the **date of the event** and a **textual description** of the failure. When possible, we manually analyze these descriptions to determine whether a failure can be associated with the antenna mast subsystem.

However, for each radar unit, only a few failure events are clearly linked to the antenna mast. This sparse and weakly labeled information makes it difficult to train supervised models for anomaly detection or health assessment.

To address this limitation, we rely on an **unsupervised approach based on autoencoders** (Lee., Jung, Song, and Choo, 2020), which allows us to learn the system's normal behavior directly from the data. Deviations from this learned behavior can then be interpreted as potential anomalies. The next section details the dataset structure, and the model architecture used in our framework.

2.3. Model Architecture

As discussed in the previous section, the lack of sufficient labeled failure data — especially for the antenna mast — makes supervised learning unsuitable. We instead adopt an **unsupervised anomaly detection approach** based on a **fully connected autoencoder (FCAE)** (Basora, Bry, Olive, and Freeman, 2021). Autoencoders are neural networks trained to reconstruct their input. When trained only on normal data, they can later identify anomalies by measuring reconstruction errors.

An autoencoder consists of two functions:

- The **encoder**, which maps an input vector $x \in \mathbf{R}^{\boldsymbol{d}}$ to a hidden (latent) representation $y \in \mathbf{R}^{\boldsymbol{h}}$, through a non-linear transformation:
$$y = g(W \cdot s + b)$$
- The **decoder**, which reconstructs the original input from the latent representation:
$$\hat{x} = g(W' \cdot y + b')$$

In our case:

- d corresponds to the input dimension after flattening a sequence (i.e. $sequence\_length \cdot n\_features$),
- h corresponds to the size of the **latent space**, i.e., the most compact representation of the sequence,
- $g(\cdot)$ is a non-linear **activation function**, used to introduce non-linearity into the model. We use **ReLU** (Rectified Linear Unit) in the hidden layers to enable the model to capture complex patterns and sparsity in the representation. A **Sigmoid** activation is used in the final layer to ensure the outputs remain within a stable numeric range $[0, 1]$, compatible with normalized input data.

The model is trained to minimize the reconstruction loss between the input x and the output $\hat{x}$, using the **mean squared error (MSE)** loss:

$$MSE(x, \hat{x}) = \frac{1}{N}\sum_{i=1}^{N} \|x_i - \hat{x_i}\|^2$$

The architecture is **symmetric**, composed of three dense layers in both encoder and decoder (Figure 1). The layer dimensions are defined as:

- $dim1 = sequence\_length \cdot n\_features$
- $dim2 = 0.7 \cdot dim1$
- $dim3 = 0.7 \cdot dim2$
- $nf = 0.7 \cdot dim3$

The full structure of the model is summarized in Table 1 and 2. This architecture allows the model to learn compact representations of normal sequences of radar sensor data. In the following section, we evaluate how well the reconstruction error can highlight deviations from expected behaviour.

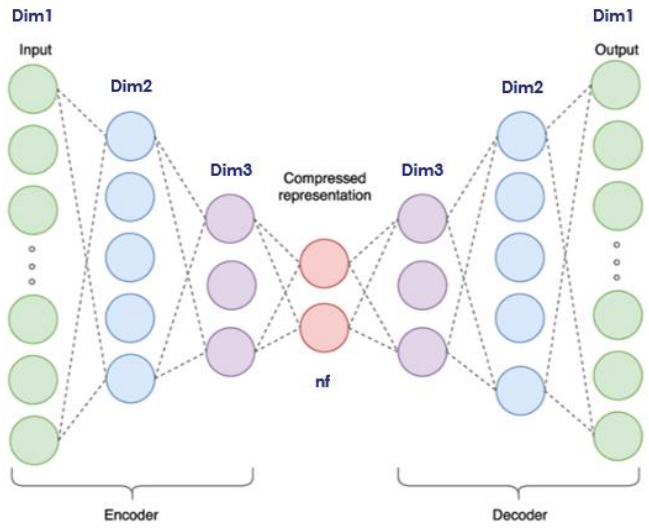

Figure 1: Fully connected autoencoder

3. SINGLE FEATURE ANALYSIS

This first stage of our framework focuses on analyzing each sensor feature individually. The goal is to understand the behavior of single variables over time, detect simple anomalies, and identify which features may be most informative.

This approach also helps validate our preprocessing pipeline and test the performance of the autoencoder in a simplified setting before moving to multi-feature analysis.

Compo-nent	Operation	Activa-tion	Role
Input	[batch_size, sequence_length, n_features] → flatten	—	Input sequence reshaped into a flat vector for dense processing.
Encoder Layer 1	Linear: dim1 → dim2	ReLu	Reduces dimensionali-ty; introduces non-linearity to learn complex relationships.
Encoder Layer 2	Linear: dim2 → dim3	ReLu	Further compression while preserving non-linear dependencies.
Encoder Layer 3	Linear: dim3 → nf	—	Projects into the latent space; no activation to preserve full representa-tional range.
Latent Space	Vector of size nf	—	Most compact form of the input; captures key features of the sequence.

Table 1 : Fully connected autoencoder architecture (1/2)

In this section, we describe the preprocessing steps, the training and testing procedure, and present the main results of this single-feature evaluation.

3.1. Data Preprocessing

For this single-feature analysis, we extract one sensor signal from radar unit "Bravo" — for example, **actuator_1_pressure**.

Compo-nent	Operation	Activa-tion	Role
Decoder Layer 1	Linear: nf → dim3	ReLu	Starts reconstruc-tion; ReLU helps recover complexity from compressed data.
Decoder Layer 2	Linear: dim3 → dim2	ReLu	Progressive reconstructio n toward the original shape.
Decoder Layer 3	Linear: dim2 → dim1	Sigmoid	Final step; sigmoid constrains output between 0 and 1 for stability.
Output	Reshaped	—	Reconstructe d sequence, aligned with the original input format.

Table 2 : Fully connected autoencoder architecture (2/2)

The preprocessing steps are as follows:

1. Remove NaN values from the dataset.
2. An outlier x is identified and removed if: $|x - \mu| < \alpha \cdot \sigma \ \ with \ \alpha$ set to a fixed value.
3. Removed values are then **interpolated** to maintain continuity in the time series.
4. The dataset is split into two parts: a **training set**, taken from a period considered representative of **normal behavior** (e.g., up to March 1, 2024, for radar "Bravo") and a **test set**, covering the remaining data, useful for evaluation on the same unit.
5. Two new columns are added:
   - **delta_timestamp**: time difference between consecutive data points,
   - **session_id**: groups data points into sessions. If the time gap between two consecutive points is less than N minutes, they belong to the same session, as shown in Figure 2.
6. The training data is scaled using a MinMaxScaler, and the same scaler is applied to the test data.
7. Within each session, we generate sequences of **20 timestamps** with an **overlap of 10**. Sequences that cannot reach 20 timestamps (typically at session boundaries) are discarded (Figure 3). This ensures that sequences are composed of temporally consistent points and avoids including data affected by sensor inactivity or radar shutdowns.
8. The final output of this process is a set of sequences, each structured as [sequence_length, n_feature]. This format matches the expected input of our autoencoder model.

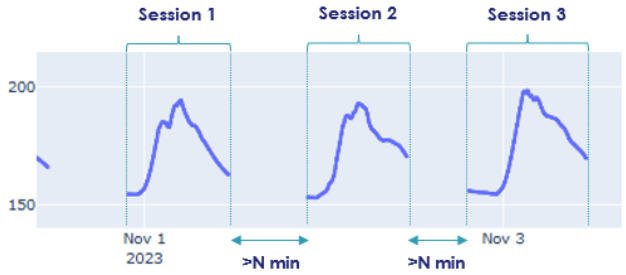

Figure 2: Session creation process

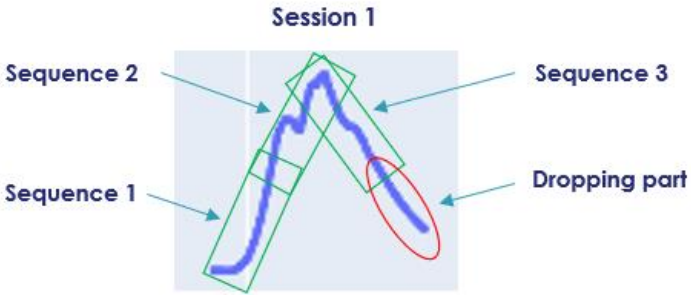

Figure 3: Sequences creation within session boundaries

3.2. Training

To train our autoencoder on the single-feature sequences, we use a **5-fold cross-validation** strategy to ensure robust performance evaluation.

The main hyperparameters are:

- Epochs: 20
- Batch size: 32
- Folds: 5 (using KFold() from scikit-learn)

Before training, the dataset is split into **training and validation folds**. For each fold, the model is trained on four subsets and validated on the fifth. This cross-validation setup reduces the risk of overfitting and improves the model's generalization ability.

Each fold is transformed into a **dataloader**, a utility that simplifies batch-wise loading and efficient iteration during model training.

The model is trained to **minimize the Mean Squared Error (MSE)** between the input and its reconstruction. This loss function penalizes larger deviations more strongly and is well-suited for regression-based anomaly detection tasks like ours.

After the five folds are completed, we retain the model configuration that achieves the lowest average validation MSE.

3.3. Anomaly Score

Once the model has been trained on the **normal data**, we must define a metric to evaluate how well a new sequence is reconstructed — and thus, to detect potential anomalies.

At this stage, we introduce the concept of an **anomaly score**, which quantifies the degree of deviation between an input sequence and its reconstruction. This score isn't used for fold selection as it relies on the **distribution of reconstruction errors** observed on the training dataset.

To account for correlations across features (in multi-feature extensions), we compute the **Mahalanobis distance** (Ahmad, Rekowski, Nedelkoski, and Ka, 2020; Jin, Ma, Cheng, and Pecht, 2012; Malhotra, Ramakrishnan, Anand, Vig, Agarwal, and Shroff, 2016) between the reconstruction error of a test sequence and the distribution of errors from the training set.

In a multivariate space, the **Mahalanobis distance** measures the distance of a point to the center of a data distribution, considering its **covariance structure** — in other words, its "shape". Unlike Euclidean distance, Mahalanobis distance penalizes deviations based on how likely they are in the context of the underlying distribution.

We calculate the **reconstruction error vectors** (i.e., the absolute difference between input and output) for all training sequences. This gives us a reference cloud of "normal" error values. From this set, we estimate the parameters of the underlying **multivariate Gaussian distribution**:

- μ: the mean of the reconstruction errors,
- Σ: the covariance matrix.

These parameters are estimated using **Maximum Likelihood Estimation (MLE)**.

Before using this Gaussian model to compute Mahalanobis distances, we verify that the distribution of reconstruction errors approximates a normal distribution. To do this, we plot the error distributions for several features (e.g., actuator_1_pressure, actuator_2_pressure, oil_tank_temperature) and visually inspect their Gaussianity.

This step is crucial, as the validity of Mahalanobis distance relies on the assumption that errors follow a multivariate Gaussian distribution. If this assumption holds, we can then compute, for each test sequence, its anomaly score as the Mahalanobis distance to the reference distribution.

As shown in Figure 4, Figure 5 and Figure 6, the reconstruction error distribution is clearly NOT Gaussian but right skewed. To eliminate the right-skewness, we apply a Box-Cox transformation to the reconstruction error values.

Box-Cox function is defined as follow:

$$f(y,\lambda) = \begin{cases} \dfrac{y^{\lambda}-1}{\lambda} & if\ \ \lambda \neq 0 \\ log(y) & if\ \ \lambda = 0 \end{cases}$$

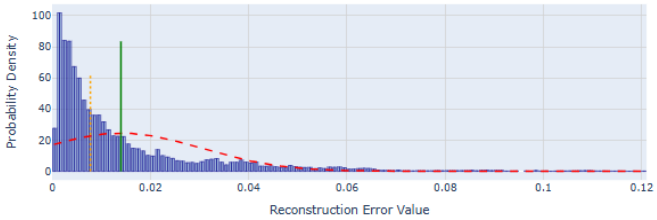

Figure 4: Reconstruction errors distribution for actuator_1_pressure

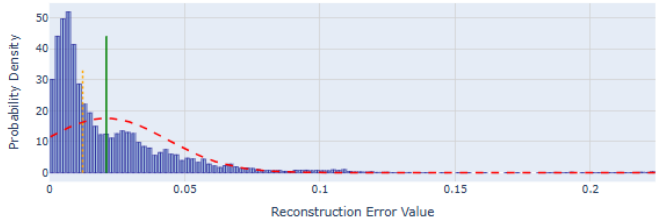

Figure 5: Reconstruction errors distribution for actuator_2_pressure

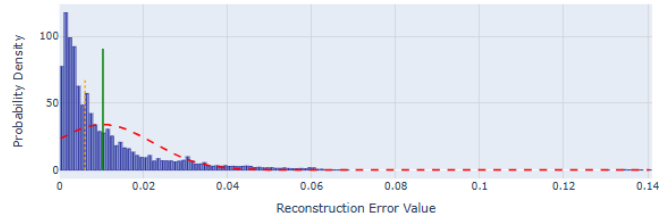

Figure 6: Reconstruction errors distribution for oil_tank_temperature

Although the reconstruction error distributions were initially right-skewed, applying a **Box-Cox transformation** allowed us to normalize them, as shown in Figures 7 – 9, making it coherent to estimate the parameters μ and Σ of the resulting Gaussian distribution:

We define **N** the **number of sequences** in the training dataset.

We define $\boldsymbol{\lambda} = (\boldsymbol{\lambda^1}, \dots, \boldsymbol{\lambda^F})$ the **Box-Cox parameters** for each feature and $f(.,\lambda): x \mapsto f(x,\lambda)$ the **Box Cox function.**

Figure 7: Box-Cox transformed reconstruction errors distribution for actuator_1_pressure

Figure 8: Box-Cox transformed reconstruction errors distribution for actuator_2_pressure

Figure 9: Box-Cox transformed reconstruction errors distribution for oil_tank_temperature

We define e_i^f the mean over time for the i^{th} sequence of the reconstruction error and for the f^{th} feature of the training dataset

We can calculate the first parameter of our Gaussian distribution:

$$\mu^f = \frac{1}{N}\sum_{i=1}^{N} f(e_i^f, \lambda^f) \quad then \quad \mu = (\mu^1, \dots, \mu^F)$$

Then, we deduce Σ the covariance matrix of this distribution:

$$\Sigma = \frac{1}{N}\sum_{i=1}^{N} (f(e_i, \lambda) - \mu)(f(e_i, \lambda) - \mu)^T$$

Then we compute the Mahalanobis score for the training dataset and normalize this score. For the i^{th} sequence:

$$M_i = \frac{(f(e_i, \lambda) - \mu)\Sigma^{-1}(f(e_i, \lambda) - \mu)^T - \min\limits_{j\in[\![1;N]\!]}(M_j)}{\max\limits_{j\in[\![1;N]\!]}(M_j) - \min\limits_{j\in[\![1;N]\!]}(M_j)}$$

3.4. Testing

To evaluate the model's performance, we define a testing procedure that mirrors the training pipeline while ensuring that no overlap occurs with the training data:

1. We extract testing data either from:
2. A different radar unit than "BRAVO" (to evaluate generalization),
3. Or from the last segment of radar "BRAVO" (e.g., the remaining part of the data), which was not used during training.
4. Remove NaN values.
5. Segment the data into sessions using the same delta_timestamp and session_id logic as in training.
6. Build sequences of 20 timestamps with an overlap of 10, as previously defined.
7. The trained autoencoder is applied to each test sequence to compute the reconstruction error vector.
8. For each sequence, to stabilize the error distribution, we apply a Box-Cox transformation $f(e_i, \lambda)$, and compute the Mahalanobis distance for each sequence i as:

$$M_i = \frac{(f(e_i, \lambda) - \mu)\Sigma^{-1}(f(e_i, \lambda) - \mu)^T - \min\limits_{j\in[\![1;N]\!]}(M_j)}{\max\limits_{j\in[\![1;N]\!]}(M_j) - \min\limits_{j\in[\![1;N]\!]}(M_j)}$$

where:

- $f(e_i, \lambda)$ is the Box-Cox transformed reconstruction error of sequence i,
- μ and Σ are the Gaussian parameters fitted on the training reconstruction errors,
- The final score M_i is normalized using training scaler.

This procedure allows us to evaluate the model's ability to detect unusual behavior in new data, based solely on its deviation from previously learned normal patterns.

3.5. Results & Interpretation

To evaluate the effectiveness of the proposed approach, we analyze the Mahalanobis scores computed on the test data. Our goal is to identify and interpret anomaly spikes, i.e., sharp increases in the anomaly score, by comparing them directly to the original sensor signal.

This analysis focuses exclusively on radar "Delta", allowing us to visually inspect how well the anomaly score aligns with potential abnormal patterns in the signal. Other radar units will be reserved for a more in-depth evaluation in the multi-feature analysis presented in the next section.

3.5.1. Actuator_1_pressure

In analyzing the Mahalanobis scores for the actuator_1_pressure feature presented in Figure 10, two distinct patterns emerge: sharp anomaly spikes and a growing trend. These patterns provide valuable insight into the underlying behavior of the system and its potential deviations.

The sharp spikes in the Mahalanobis scores are particularly noteworthy, as they indicate abrupt shifts in the system's behavior. To better understand these anomalies, we focus on two specific instances where these spikes occur: June 19, 2023 (Figure 11), and August 16, 2023 (Figure 12). By comparing these spikes with the original sensor signal, we gain a clearer view of the signal's state at those moments, as well as the reconstructed signal provided by the model at the same time.

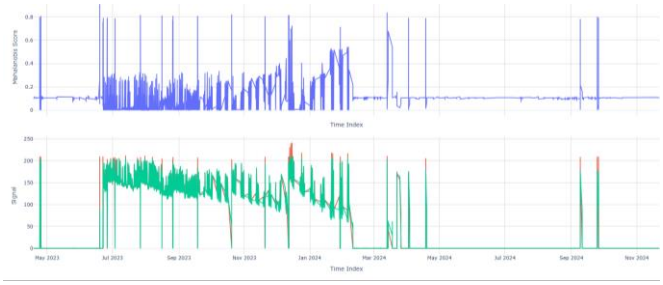

Figure 10: Mahalanobis score and reconstructed signal for actuator_1_pressure

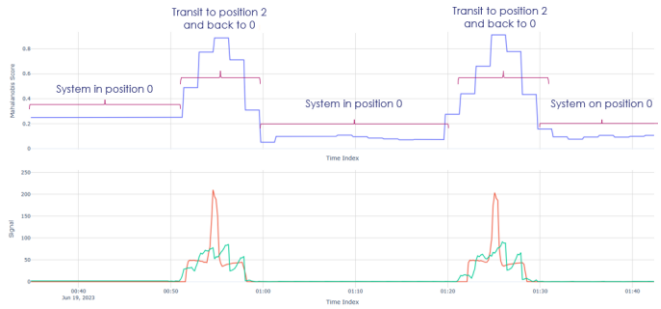

Figure 11: Mahalanobis spikes analysis on June 19, 2023 on "Delta" radar

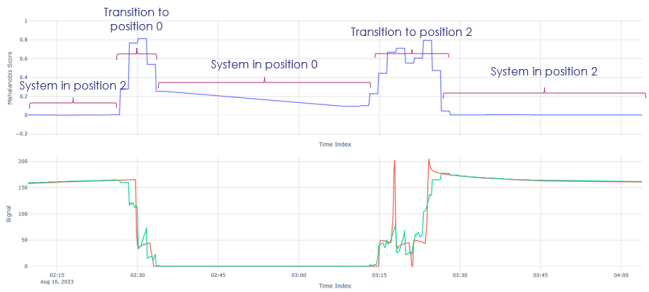

Figure 12: Mahalanobis spikes analysis on August 16, 2023, on "DELTA" radar

Upon closer inspection of the two anomaly spikes, we observe that both June 19, 2023, and August 16, 2023, exhibit similar behavior. These dates correspond to transition moments, where the antenna mast shifts from one position to another. Such abrupt transitions are particularly challenging for the model to interpret, as they are characterized by more violent changes compared to a stable, stationary phase. These sharp transitions introduce complexity into the model's ability to effectively capture and predict the system's behavior during these moments.

The rapid and unpredictable nature of these transitions creates a signal pattern that deviates significantly from the typical behavior the model was trained on. This can lead to higher Mahalanobis scores, as the model may not fully comprehend the dynamics of such drastic movements. However, upon further review of the maintenance data, it appears that these rapid transition phases do not seem to correlate with any reported failures by the maintenance team. This lack of correlation suggests that the system might be designed to withstand such transition behaviors without resulting in failure, despite the apparent disruption they cause in the sensor signals.

In conclusion, while the Mahalanobis score spikes on these specific dates indicate significant signal shifts, the underlying cause appears to be the system's transition from one state to another, rather than any underlying failure. This observation is further supported by the maintenance records, which do not show any related failures, highlighting the resilience of the system during these transition periods.

3.5.2. Actuator_2_pressure

The analysis of actuator_2_pressure reveals that spikes in the Mahalanobis score are highly correlated with those observed in actuator_1_pressure (Figure 13 and Figure 14). These spikes appear to be caused by similar transition phases, during which both actuators experience significant shifts in their behavior. However, the decreasing trend observed in actuator_1_pressure is not present in actuator_2_pressure.

Figure 13: Mahalanobis score for actuator_2_pressure on "DELTA"

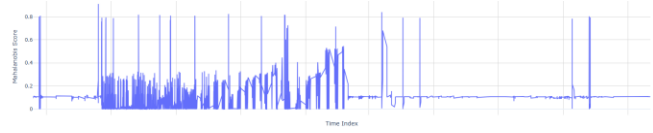

Figure 14: Mahalanobis score for actuator_1_pressure on "DELTA"

Given the symmetry between the two actuators in their role within the radar subsystem, it is not surprising that the Mahalanobis score spikes for actuator_1_pressure and actuator_2_pressure are closely aligned. The correlation between these two features is expected, as both actuators typically operate in tandem. Instead of focusing on the individual spikes in each feature, it is more relevant to examine the correlation between actuator_1_pressure and actuator_2_pressure across all radar units ("ALPHA",

"BRAVO", "CHARLIE", and "DELTA"). This approach will provide deeper insights into the system's overall performance and the potential causes behind the observed anomalies.

We performed a linear regression analysis of actuator_2_pressure against actuator_1_pressure across four different radar units. The results reveal a clear deviation in the behavior of actuator_1_pressure relative to actuator_2_pressure specifically on radar "DELTA" (Figure 15 – 18).

While a strong linear relationship is observed between the two features on the other radar units ("ALPHA", "BRAVO", and "CHARLIE"), the relationship between actuator_1_pressure and actuator_2_pressure on radar "DELTA" appears much less pronounced. This deviation suggests that something unusual is occurring with the actuator pressures on "DELTA", and further investigation is required to understand the underlying cause of this discrepancy.

3.5.3. Actuator_3_pressure

Actuator_3_pressure behaves differently across the various radar units, which leads to inconsistencies when the signal is reconstructed. The model, which is trained to reconstruct the signal based on data from radar "BRAVO", struggles to provide a meaningful reconstruction when applied to other radar units, as the underlying behavior of actuator_3_pressure differs significantly from one radar to another.

This discrepancy suggests that the model may not fully capture the unique operating conditions of actuator_3 across all radars. A deeper understanding of the specific context and operation of actuator_3 is needed to interpret why these variations occur and how they affect the reconstructed signal. Further investigation into the operating conditions and the behavior of actuator_3 across the radar units will help clarify this issue.

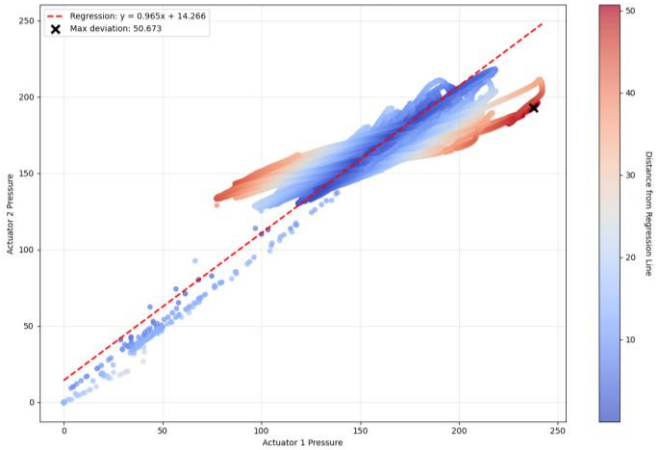

Figure 15: Linear Regression between actuator_1_pressure and actuator_2_pressure on "DELTA"

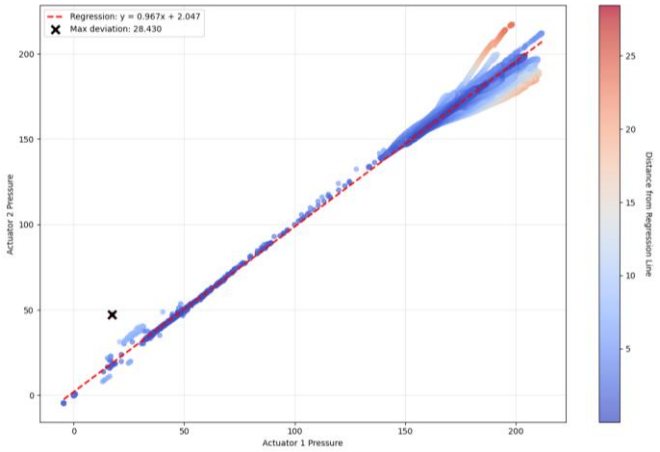

Figure 16: Linear Regression between actuator_1_pressure and actuator_2_pressure on "BRAVO"

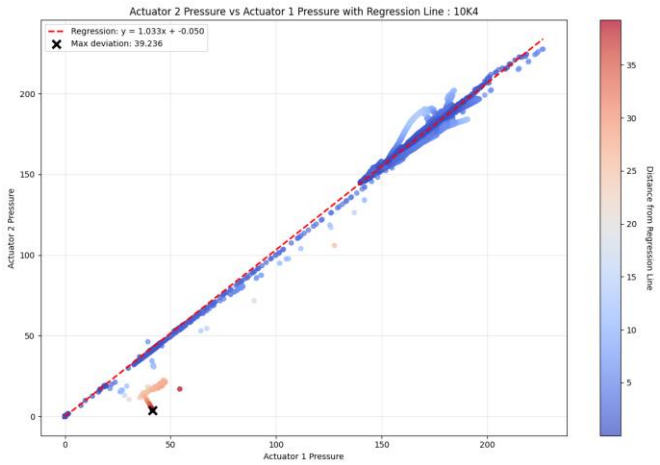

Figure 17: Linear Regression between actuator_1_pressure and actuator_2_pressure on "ALPHA"

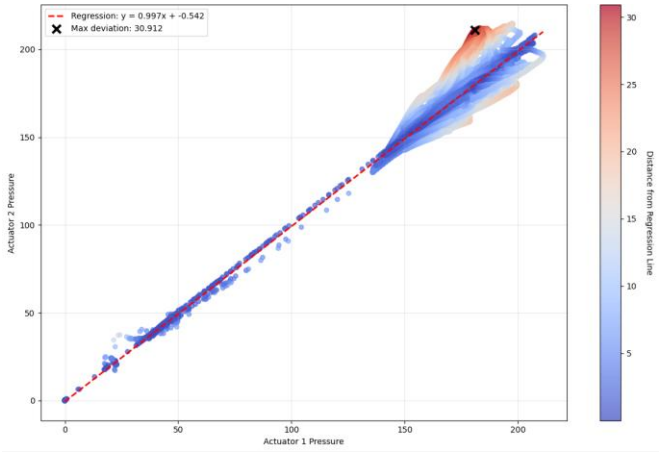

Figure 18: Linear Regression between actuator_1_pressure and actuator_2_pressure on "CHARLIE"

3.5.4. Oil_tank_temperature

The Mahalanobis score for the oil_tank_temperature feature reveals a single pronounced spike (Figure 19), which can be directly linked to an unusually high value in the signal — a temperature close to 110°C, compared to the normal operating range around 40°C. In this case, the analysis is straightforward: the model effectively flags this outlier, indicating that it is clearly capable of identifying significant deviations in the signal. This spike suggests an anomaly in the oil tank temperature, and the model's ability to detect such outliers demonstrates its robustness in recognizing abnormal conditions.

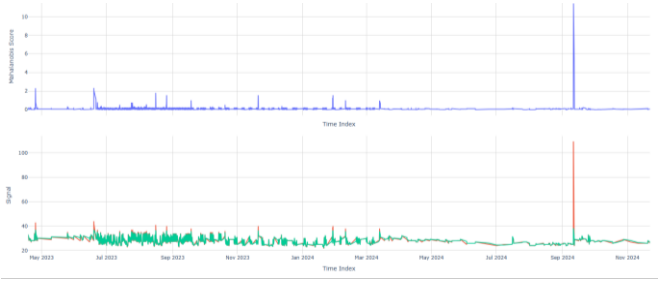

Figure 19: Mahalanobis score and reconstructed signal for oil_tank_temperature

3.5.5. Conclusion

In this initial stage of our framework, we have analyzed individual sensor features to better understand their behavior over time and detect simple anomalies. Our evaluation of actuator_1_pressure revealed both transition-related anomalies and an unexpected decreasing trend. Actuator_2_pressure showed a clear correlation with actuator_1_pressure across most radar units, except for "DELTA", where this correlation was less apparent. Actuator_3_pressure exhibited distinct behaviors between radar units, complicating the interpretation of whether the observed variations are normal, it will require further investigations. Finally, the oil_tank_temperature feature flagged a clear outlier, which was easily identified by the model.

This deeper understanding of individual sensor behaviors lays the foundation for our next step: a multi-feature analysis. By combining multiple features, we aim to capture more complex interactions and refine the detection of anomalies, ultimately improving the overall performance of our anomaly detection framework.

4. MULTI-FEATURES ANALYSIS

In this phase of the analysis, we expand the scope to consider multiple sensors features simultaneously, aiming to capture more complex relationships and interactions between the features. The global process remains like the single-feature analysis, but with a few important adjustments to improve the model's accuracy and relevance.

4.1. Process Flow

From the previous analysis, we observed that actuator_3_pressure behaves differently between radar units "DELTA" and "BRAVO", leading to a poorly reconstructed signal. As a result, we decided not to include this feature in the global analysis. Additionally, instead of extracting the entire dataset, we filter the data based on the radar's position. We identified that the model flagged anomalies during the transition phase, but these were not related to actual system alerts. Therefore, we focus only on the data collected when the radar is in its deployed position (position 2).

The preprocessing steps remain the same as in the single-feature analysis. The goal is to prepare the data by handling NaN values, scaling, creating sessions and sequences and ensuring it is in the right format for the model.

We use the same model architecture as in the single-feature analysis. The model is based on an autoencoder framework, where the encoder compresses the input data and the decoder reconstructs it to detect anomalies by comparing the reconstruction error.

For the training process, we increase the number of epochs to 40 to ensure the model learns the features sufficiently. The training will be performed on the first part of the "BRAVO" dataset, using a 5-fold cross-validation technique.

Hyperparameters:

- Epochs: 40
- Batch size: 32
- Folds: 5

Once the model is trained, we compute the Mahalanobis distance parameters (Σ and μ).

The model will be tested on several radar units: "ALPHA", "BRAVO" (last part), "CHARLIE", and "DELTA". This allows us to evaluate the generalization performance of the model across different radar units.

Finally, we compute the Mahalanobis score for each radar unit. When dealing with multi-feature analysis, it's not always straightforward to determine where the fault originates: Is it a combination of all features, or is it a specific feature causing the issue? To address this, we need to break down the Mahalanobis score to evaluate the contribution of each individual feature.

We define a contribution matrix $C \in R^{F*F}$ that captures the interaction between all features. Each element represents the contribution of feature f and feature l to the total Mahalanobis score. This is calculated as:

$$\boldsymbol{C}_{f,l} = \left(\boldsymbol{e}_i^{\boldsymbol{f}} - \boldsymbol{\mu}^f\right)\boldsymbol{\Sigma}_{l,f}^{-1}\left(\boldsymbol{e}_i^{\boldsymbol{l}} - \boldsymbol{\mu}^l\right)^T$$

By definition:

$$M_i = \sum_{f=1}^{F} \sum_{l=1}^{F} C_{f,l}$$

This contribution matrix helps us understand the pairwise relationships between features in terms of their contribution to the Mahalanobis score by analyzing this matrix, we can determine whether the anomaly is driven by a single feature or a combination of features, providing a clearer understanding of the source of the issue.

4.2. Results and Interpretation

In this section, we present the results of the multi-feature analysis by plotting the Mahalanobis score alongside the original and reconstructed signals for each feature, for each radar unit. This comprehensive approach allows us to gain a complete overview of the model's performance and better understand the relationship between the original and reconstructed signals.

4.2.1. "DELTA"

For radar unit "DELTA", we observed the following key results (Figure 20):

- **No Spikes in the Mahalanobis Score**: The spikes corresponding to the transition phase, which were previously observed, no longer appear in the analysis. This indicates that by focusing only on the data collected when the radar is in its deployed position (position 2), we have successfully filtered out the anomalies caused by the transition phase.
- **Problem in actuator_1_pressure**: The primary issue with radar "DELTA" stems from the actuator_1_pressure feature, which displays a decreasing trend in the signal. This abnormal behavior, unlike what is observed in other radar units, contributes to the high Mahalanobis score and signals an anomaly in the actuator's performance.

During our analysis, we highlighted an abnormal decreasing pattern in actuator_1_pressure for radar unit "DELTA". This anomaly was found to be related to issues with the DRS/MCF, which correspond to real system failures, as in Figure 21.

Upon further investigation, we found that this abnormal behavior was noted in the DRS record sheet, specifically in the intervention section. The following entry was made:

"*Abnormal pressure was found on the telescopic pressure which may cause the mast failure. […] Require investigating the drastic drop of hydraulic pressure for both telescopic and drive actuators.*" – DRS 62213982 record sheet

This aligns with the decreasing trend observed in the Mahalanobis score, confirming that the anomaly flagged by the model corresponds to a real failure event, which could result in system malfunction if not addressed.

The increasing Mahalanobis score suggests that condition-based monitoring could be an effective method for detecting such anomalies in future cases. By continuously tracking changes in the Mahalanobis score, it may be possible to identify potential issues early and trigger maintenance actions before a failure occurs.

To understand the specific contribution of each feature to the high Mahalanobis score for radar unit "DELTA", we performed a breakdown of the Mahalanobis score using the contribution matrix for the sequence with the highest score.

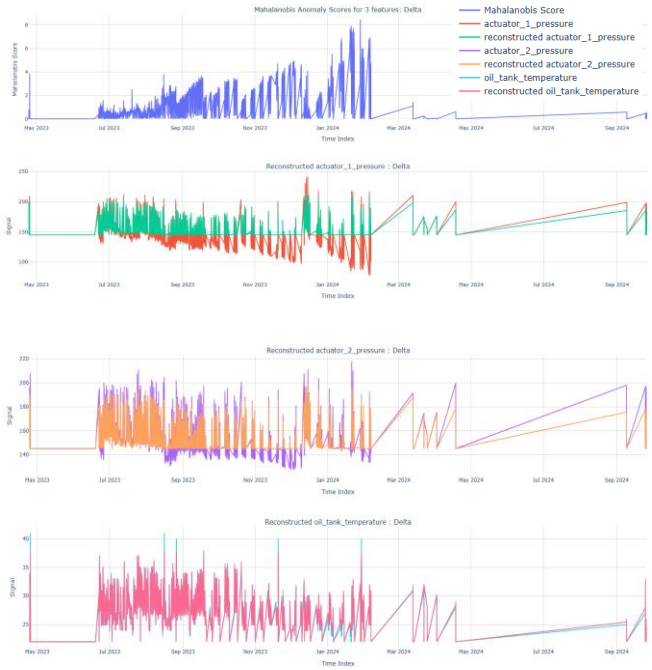

Figure 20: Mahalanobis score, signals and all reconstructed signals for "DELTA"

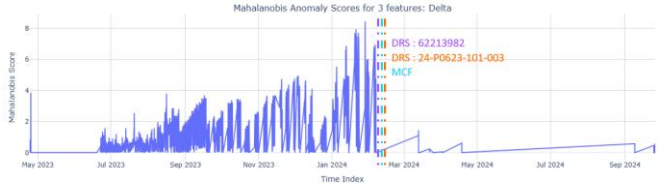

Figure 21: Mahalanobis score and alerts for "DELTA"

The analysis revealed the following key findings:

- **Actuator_1_pressure**: This feature is clearly the primary source of the anomaly. The contribution from actuator_1_pressure is significant, and its abnormal decreasing trend is driving the high Mahalanobis score.
- **Oil_tank_temperature**: In addition to actuator_1_pressure, oil_tank_temperature also appears to be problematic. While its contribution is not as

pronounced as actuator_1_pressure, it still plays a role in the overall deviation of the system from normal behavior.

This contribution matrix (Figure 22) allows us to pinpoint the specific features that contribute most to the anomalies, providing a clearer understanding of the source of the problem and guiding the next steps for further investigation.

4.2.2. "CHARLIE"

For radar unit "CHARLIE", we observed the following (Figure 23):

- **Outlier**: At first glance, the Mahalanobis score shows a single pronounced spike, which clearly flags an outlier in the oil_tank_temperature feature. This spike indicates a significant deviation from normal behavior, suggesting an anomaly.
- **High Oil Tank Temperature**: This behavior is not considered "normal," as it coincides with a bite alert indicating a high oil tank temperature issue. The alignment of the Mahalanobis spike with the alert reinforces the model's ability to flag actual system issues.

The anomaly detected in the oil_tank_temperature feature in "CHARLIE" suggests a potential issue that should be further investigated, as the model's output corresponds with the alert triggered by the system (Figure 24).

This anomaly was found to be related to issues with the DRS/MCF, which correspond to real system failures.

Upon further investigation, we found that this abnormal behavior was noted in the DRS and MCF as a sensor fault (Figure 25). The feature contribution analysis has also validated our point: Oil_tank_temperature is clearly problematic, and this result is coherent with what we've seen in the signal.

4.2.3. "BRAVO"

For radar unit "BRAVO", we observed the following (Figure 26):

- **Outlier**: At first glance, the Mahalanobis score shows a single pronounced spike, which clearly flags an outlier in the oil_tank_temperature feature. This spike indicates a significant deviation from normal behavior, suggesting an anomaly.In reality, this outlier is clearly due to sensor default as it displayed a temperature above 60k degrees Celsius.
- **Scaling challenge:** This outlier is very far from the usual range of values, which results in the Mahalanobis score also being very high at this level. Therefore, due to scale issues, the rest of the scores become barely discernible, so we decide to zoom in to overcome these scale problems:

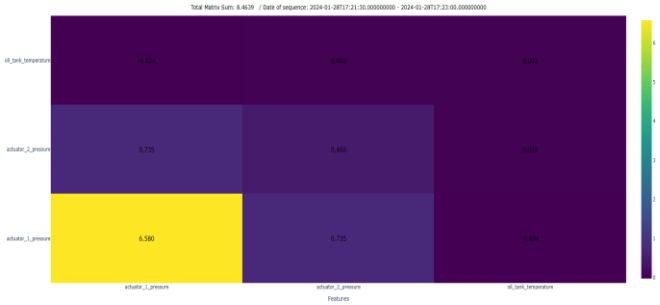

Figure 22: Mahalanobis breakdown matrix from sequence on 28/01/2024 for "DELTA"

Figure 23: Mahalanobis score, signals and all reconstructed signals for "CHARLIE"

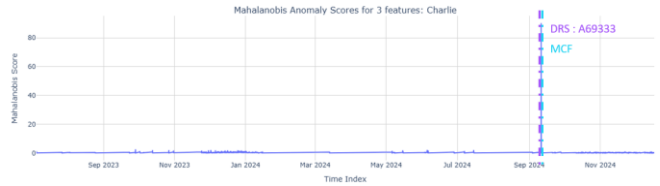

Figure 24: Mahalanobis score and alerts for "CHARLIE"

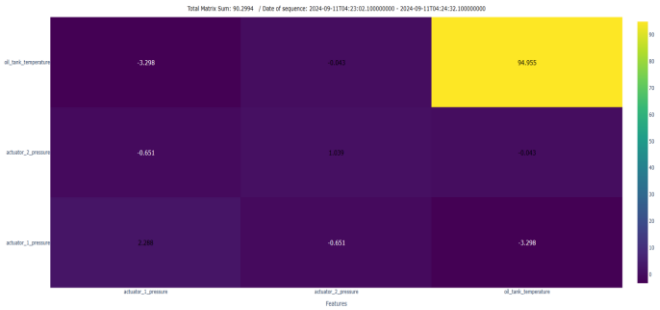

Figure 25: Mahalanobis breakdown matrix from sequence on 11/09/2024 for "CHARLIE"

Most of the anomalies flagged by the model are found to correlate with existing DRS/MCF alerts and BITE alerts (Figure 27). However, the model also detects a spike that does not appear to be associated with any known alert. For DRS A69333, the failure description indicates operation ranges exceeding radar limits:

Figure 26: Mahalanobis score, signals and all reconstructed signals for "BRAVO"

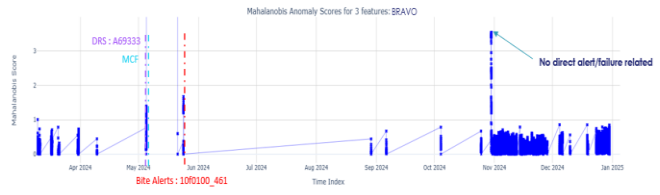

Figure 27: Zoomed Mahalanobis score and alerts for "BRAVO"

"MAST: OIL LOW TEMPERATURE FAILURE; MAST: ACTUATOR 3 PRESSURE OUT OF RANGE; MAST: ACTUATOR 2 PRESSURE OUT OF RANGE; MAST: ACTUATOR 1 PRESSURE OUT OF RANGE." The Bite alert 10f0100_461 is less straightforward and refers to an "ANTENNA DELTA PRESSURE FAULT." The analysis of feature contribution highlights the specific features responsible for each alert, providing better interpretability and insights into the model's decision-making process :

The first matrix (Figure 28) clearly highlights an outlier in the oil_tank_temperature feature, with a value exceeding 60,000°C. This result is consistent with the anomaly observed in the raw signal.

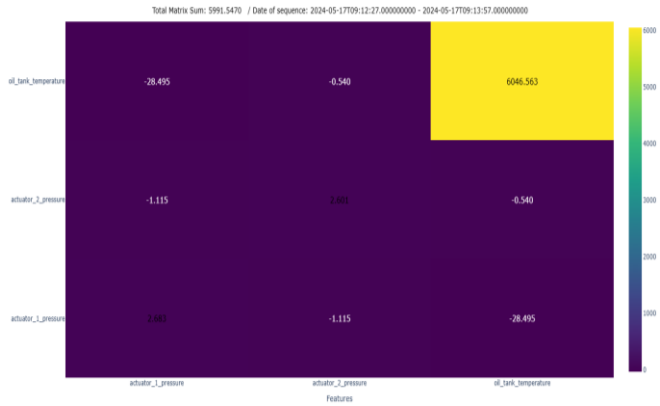

Figure 28: Mahalanobis breakdown matrix from sequence on 17/05/2024 for "BRAVO"

The second matrix (Figure 29) corresponds to the DRS and MCF alerts, which reference features operating out of their expected range. In this case, the matrix indicates that multiple features contribute to the anomaly. A closer inspection of the signal reveals a significantly high value for actuator_2_pressure, creating a noticeable pressure difference with actuator_1_pressure. In contrast, oil_tank_temperature does not appear abnormal when examined directly in the signal.

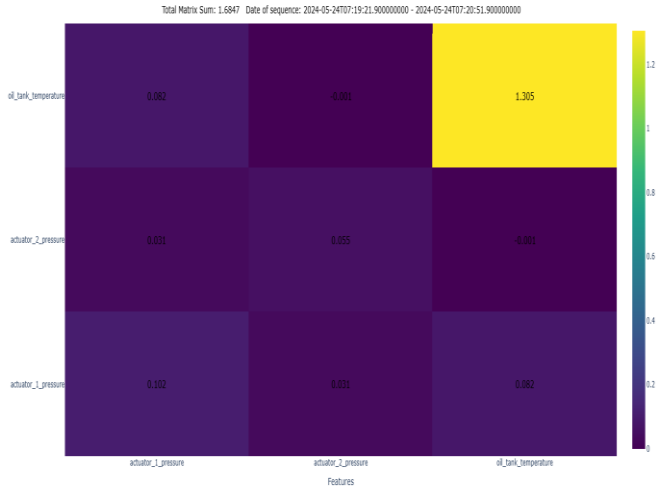

Figure 29: Mahalanobis breakdown matrix from sequence on 24/05/2024 for "BRAVO"

Finally, the third matrix (Figure 30) relates to the BITE alert labeled "ANTENNA DELTA PRESSURE FAULT." Here, oil_tank_temperature appears as a significant contributing feature. However, the link between this variable and the alert

is not straightforward. The correlation between the oil_tank_temperature anomaly and the antenna delta pressure fault remains unclear, highlighting the need for further investigation.

4.2.4. "ALPHA"

For radar unit "ALPHA", the Mahalanobis score is significantly lower compared to the scores observed for "DELTA" and "CHARLIE". No anomalies or outliers stand out in these radar units, indicating that the system's behavior is relatively stable and within expected parameters.

This radar unit exhibits normal signal patterns without any notable deviations, suggesting that no major issues are detected for these units in this analysis.

5. CONCLUSION

We have developed a model capable of flagging outliers and abnormal patterns, evaluating them through the Mahalanobis score. While the model proves effective in identifying anomalies, the next step is to determine an appropriate threshold to fully automate the anomaly detection system. This remains a challenge due to the lack of clearly faulty data, such as DRS or MCF, which makes it difficult to define a precise threshold based on observable failures.

One of the key considerations in our approach is the filtering of data based solely on the radar's deployed position (position 2). This filtering excludes data from the transition phase and when the radar is in position 0, which may result in the loss of valuable information and potentially missing anomalies. This decision raises the question of whether we should develop a model to handle these cases, as filtering restricts the scope of data available for analysis. Additionally, the model faces challenges in characterizing what constitutes an anomaly. The difficulty arises from the variety of anomaly types, including outliers and abnormal patterns, which are not always easy to define. The absence of clearly defined failure data, such as DRS or MCF, further complicates the detection process.

Future exploration should focus on testing the model with other sub-systems. This would provide a broader understanding of the system's behavior, but it is essential to consider the relationships between sub-systems, especially since the data sampling rates differ across them. Furthermore, exploring alternative models and fine-tuning their hyperparameters could lead to improved performance, although this remains challenging given the uncertainty in defining anomalies. Lastly, while the current model only uses numerical features, incorporating binary features could add significant value. By integrating binary data, we could enhance the model's ability to detect anomalies, as this type of feature may provide useful additional insights.

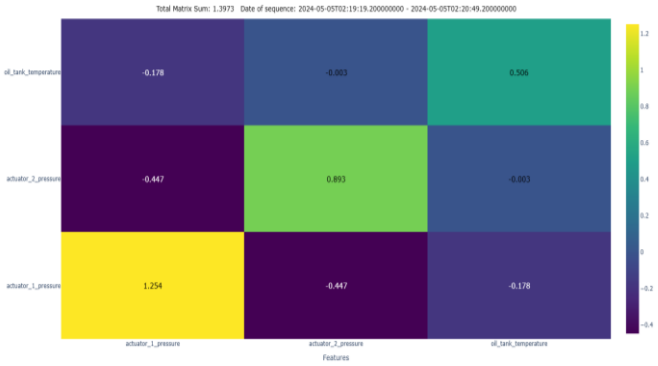

Figure 30: Mahalanobis breakdown matrix from sequence on 05/05/2024 for "BRAVO"

REFERENCES

Ahmad, S., Rekowski, K. S., Nedelkoski, S., & Ka, O. (2020). Autoencoder-based Condition Monitoring and Anomaly Detection Method for Rotating Machines. *IEEE International Conference on Big Data (Big Data)*. doi: 10.1109/BigData50022.2020.9378015

Basora, L., Bry, P., Olive, X., & Freeman, F (2021). Aircraft Fleet Health Monitoring with Anomaly Detection Techniques. *Aerospace*, vol. 8, no. 4, p. 103. doi:10.3390/ aerospace8040103

Jin, X., Ma, E. W. M., Cheng L. L., & Pecht, M. (2012). Health Monitoring of Cooling Fans Based on Mahalanobis Distance With mRMR Feature Selection. *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 8. doi: 10.1109/TIM.2012.2187240

Lee, G., Jung, M., Song, M., & Choo, J. (2020). Unsupervised anomaly detection of the gas turbine operation via convolutional auto-encode. *IEEE International Conference on Prognostics and Health Management (ICPHM)*. doi: 10.1109/ICPHM49022.2020.9187054

Leukel, J., Gonzalez, J., & Riekert, M. (2021). Adoption of machine learning technology for failure prediction in industrial maintenance: A systematic review. *Journal of Manufacturing Systems*, vol. 61, pp. 87–96. doi:10.1016/j.jmsy.2021.08.012

Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., & Shroff, G. (2016). LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection. doi: 10.48550/arXiv.1607.00148