RAVEN: Unsupervised Anomaly Detection in Multivariate Jet Engine Time Series using Residual Learning on Real Test Data

Nouf Almesafri, Mohamed Ragab, Salama AlMheiri, Zahi Mohamed, Abdulla Alseiari

Propulsion and Space Research Center, Technology Innovation Institute, Abu Dhabi, United Arab Emirates

Nouf.almesafri@tii.ae
Mohamed.adam@tii.ae
Salama.almheiri@tii.ae
Zahi.mohamed@tii.ae
Abdulla.alseiari@tii.ae

ABSTRACT

Jet engines operate under demanding conditions, subjecting critical components to gradual wear and degradation over time. Early identification of incipient faults is essential for maintaining performance, safety, and reliability. Detecting incipient faults early is essential but remains difficult due to two major challenges: the scarcity of faulty data and the strong variability in operating conditions that obscure fault signatures. Most existing anomaly detection approaches rely on simulated datasets or assume the availability of labeled faults, limiting their applicability to real-world engine monitoring. In this work, we introduce RAVEN, a fully unsupervised anomaly detection framework designed for jet engine monitoring under real test conditions. RAVEN integrates (i) a regression-based residual model to normalize sensor responses against varying operating regimes, with (ii) a deep LSTM autoencoder that captures subtle deviations in time-series behavior without requiring fault labels. By explicitly addressing operational variability, sensor noise, and label scarcity, RAVEN provides a robust pathway for early fault detection. We validate RAVEN on real jet engine test data, demonstrating its ability to detect anomalies under diverse operating conditions. Results show that our approach delivers reliable detection performance in scenarios where conventional approaches struggle, offering a practical and scalable solution for propulsion system health monitoring.

1. INTRODUCTION

Jet engines play a crucial role in modern aviation, powering commercial aircraft, military jets, and various aerospace applications Talebi et al. (2025). To generate the necessary thrust for sustained flight, they operate under extreme temperatures, pressures, and rotational speeds, which gradually degrade internal components such as turbines, compressors, and bearings. Over time, this degradation can lead to a drop in performance, increased fuel consumption, increased emissions, and reduced flight safety Miao et al. (2024). To identify early faults, anomaly detection is employed. It refers to the analytical process of detecting irregular patterns or events within a dataset and examining the conditions under which they occur. Anomaly detection significance has increased as the scale of modern data renders manual identification impractical, necessitating automated solutions. Such methods have been successfully applied in cybersecurity intrusion detection (Ahmed, Mahmood, & Hu, 2016), medical diagnostics such as ECG analysis (Fernando, Gammulle, Denman, Sridharan, & Fookes, 2021), predictive maintenance in manufacturing (Davari, Veloso, Ribeiro, Pereira, & Gama, 2021), financial market risk monitoring (Hodge & Austin, 2004), and environmental monitoring using sensor data (Hill & Minsker, 2010). One of the core challenges lies in the unpredictable nature of anomalies, which often limits the use of conventional machine learning approaches that require labeled data sets, particularly in time series contexts (Bahri, Salutari, Putina, & Sozio, 2022).

Traditionally, anomaly detection in jet engines has relied on statistical and rule-based methods, which monitor sensor readings and flag deviations from predefined thresholds or expected patterns (Wong, Leckie, & Ramamohanarao, 2002). While these conventional approaches are straightforward, they often struggle to capture complex, nonlinear relationships in high-dimensional data, leading to false alarms or missed anomalies. To overcome these limitations, AI-based anomaly detection systems have gained increasing attention Huang et al. (2025). By learning patterns from sensor data during healthy operation, these systems can identify subtle deviations indicative of developing faults, enabling predictive maintenance, improved reliability, and more efficient operation (Kurz et al., 2008). Machine learning techniques such as Isolation Forest, One-Class SVM, and PCA have been successfully applied for anomaly detection,

while deep learning approaches, including autoencoders and recurrent neural networks, have shown superior performance in capturing complex temporal and spatial patterns in engine sensor data (Kucuk & Uysal, 2022). Figure 1 illustrates the overview of anomaly detection in jet engines.

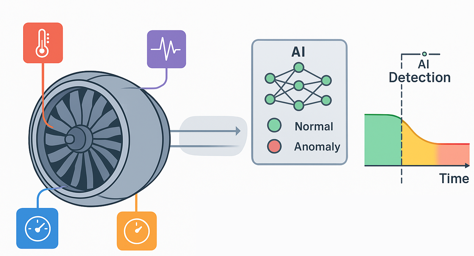

Figure 1: Overview of AI-powered anomaly detection in jet engines. Multiple sensors monitor engine components, feeding data to machine learning systems that detect early signs of degradation before critical failures occur.

A key factor in these AI-based approaches is that the underlying data are often time-series measurements Zamanzadeh Darban et al. (2024); sequences of readings recorded over time, where the temporal order carries critical information. In engineering and industrial applications, such data is collected from sensors that measure parameters such as vibration, temperature, pressure, and rotational speed (RPM). These measurements are critical for monitoring system performance, diagnosing faults, and enabling predictive maintenance in complex machinery such as turbines, engines, and manufacturing equipment (Tang, Yuan, & Zhu, 2019). The inherent temporal dependencies in time-series data allow analysts to detect trends, seasonal patterns, and sudden deviations, which can indicate abnormal or unsafe operating conditions (Box, 2013). However, analyzing sensor-based time-series data poses challenges due to noise, high dimensionality, and variability in operating conditions, emphasizing the need for advanced analytical techniques such as statistical modeling, signal processing, and machine learning.

Time-series data in the context of anomaly detection can be either labeled or unlabeled, which determines the type of technique applied. Anomaly detection techniques can be broadly categorized into supervised, unsupervised, and semi-supervised approaches, each differing in their reliance on labeled data. Supervised anomaly detection requires datasets labeled as “normal” or “anomalous” to train a model that can classify new instances accurately (Görnitz, Kloft, Rieck, & Brefeld, 2013). While this method can achieve high precision, it is often limited by the scarcity and high cost of obtaining labeled anomaly data. Unsupervised anomaly detection, on the other hand, does not require labels and instead relies on the assumption that anomalies are rare and significantly different from normal patterns (Meng et al., 2019). This makes it well-suited for real-world scenarios where anomalies are diverse and unpredictable. Semi-supervised anomaly detection serves as a middle ground, leveraging a training set consisting primarily or entirely of normal data to build a model that flags deviations as potential anomalies (Akcay, Atapour-Abarghouei, & Breckon, 2018). The choice among these methods depends on factors such as data availability, labeling costs, and the variability of normal and abnormal patterns in the target domain.

When applied to time-series sensor data, such as vibration, temperature, pressure, or rotational speed measurements, anomaly detection presents additional challenges. In particular, variability in operating conditions such as changes in load, speed, or environmental factors can significantly affect sensor readings, making it difficult for conventional models to distinguish between normal variations and true anomalies. Moreover, the temporal dependencies must be preserved to correctly capture evolving patterns and subtle deviations. Traditional models often fail to exploit this sequential structure effectively. For such cases, models with memory capabilities, like the Long Short-Term Memory (LSTM) network, are particularly advantageous, as they can retain information from earlier time steps, enabling the detection of anomalies that emerge gradually or depend on long-term temporal context (Du et al., 2017; Malhotra, 2016; Park et al., 2018; Zhou et al., 2020).

In this work, we propose a framework for anomaly detection in jet engines that integrates regression residual modeling with an autoencoder-based reconstruction stage. Prior studies have applied reconstruction methods or prediction models directly on raw sensor data, which often struggle to distinguish genuine faults from variations in operating conditions (Malhotra et al., 2016; Wei et al., 2023; Wang & Tong, 2022; Kieu et al., 2022). Our framework addresses this by explicitly separating operational condition sensors (e.g., fuel flow) from response sensors (e.g., Tt9, Pt3), first using an LSTM regression model to predict responses and then analyzing the resulting residuals with an LSTM autoencoder. This residual–autoencoder design allows anomalies to be detected in a way that accounts for operational variability. Importantly, we evaluate the framework at the event level on real jet engine test data, providing a practical demonstration of its effectiveness in contrast to simulation-based or component-level studies commonly found in literature.

1.1. Contributions

The main contributions are as follows:

- RAVEN – Robust Anomaly Detection under Variable Engine Conditions: A novel residual-based framework that employs a two-step residual learning approach, combining regression and autoencoding of residuals, for unsupervised anomaly detection.

Commented [MR1]: Highlight the issue of variability of working condition as challenge not solved by the existing approaches

Commented [NM2R1]: Addressed.

- Use of real-world test data: The study employs sensor measurements collected from an operational system, ensuring that the results are representative of practical conditions rather than simulated environments.
- Consideration of operational variability: The analysis incorporates variations in operating parameters to reflect realistic system behavior.
- Healthy-data-only training: The proposed model is trained exclusively on healthy time-series data, enabling effective unsupervised anomaly detection.
- Identification of real anomaly indicators: The method is capable of extracting meaningful features that serve as indicators of actual system anomalies, supporting both early fault detection and root cause analysis.

2. OUR APPROACH

The proposed framework (Figure 2) consists of two main components: an LSTM-based regression model and an LSTM-based autoencoder. Unlike conventional anomaly detection methods that treat sensor data uniformly, our framework explicitly distinguishes between operational condition sensors (e.g., fuel flow) and response sensors (e.g., Tt9, Pt3). The regression model first predicts response sensors from operational conditions, isolating residuals that capture deviations from expected engine behavior. These residuals are then passed to the autoencoder, which is trained to reconstruct only healthy residual patterns. Consequently, faulty conditions yield high reconstruction errors that can be flagged via a threshold. This residual–autoencoder pipeline allows anomalies to be detected in a way that accounts for engine operational variability, and our evaluation is conducted at the event level on real jet engine test data, demonstrating the framework's practical applicability beyond simulation-based studies.

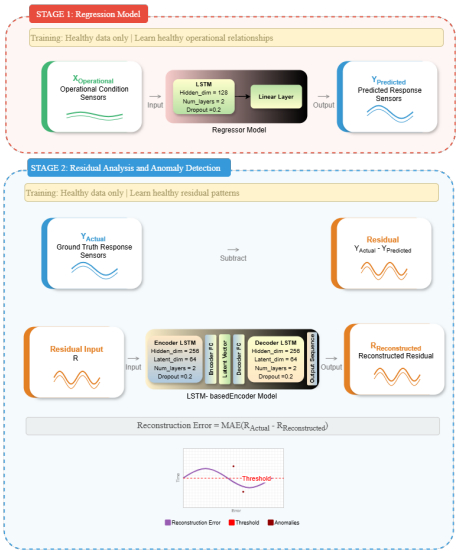

Figure 2. Overview of the proposed anomaly detection pipeline. Operational indicator sensors are fed into a regression model to predict response sensor behavior. The residual between predicted and actual readings is input to an LSTM autoencoder, which reconstructs the residual signals and detects anomalies when reconstruction error exceeds a threshold.

2.1. Regression model

The regression model takes the operational indicator sensors X_{op} as input and predicts the response sensors. The representation of the operational sensors is given in Eq. (1), where m denotes the number of operational sensors and T the number of time steps in the experiment:

$$X_{op} = \{x_t^{(j)} | \, j = 1,2,\dots,m; t = 1,2,\dots,T\} \quad (1)$$

The operational data is scaled to $[0,1]$ using MinMax scaling. For sensor j, the transformation is given in Eq. (2):

$$x_t^{(j)} = \frac{x_t^{(j)} - min(x^{(j)})}{max(x^{(j)}) - min(x^{(j)})} \quad (2)$$

where $min(x^{(j)})$ and $max(x^{(j)})$ denote the minimum and maximum values of sensor j across the training set.

To capture the sequential nature of the data, we use a sliding window with a fixed length $W = 1000$ samples and a stride $S = 500$. Each windowed input, denoted as $X_{op}^{(i)}$ for window

number i, contains all m operational sensors over W consecutive time steps, as defined in Eq. (3):

$$X_{op}^{(i)} = \{x_t^{(j)} \mid \mathrm{j} = 1,2,\ldots,m;\ \mathrm{t} = (\mathrm{i}-1)\mathrm{S}+1,\ldots,(\mathrm{i}-1)\mathrm{S}+\mathrm{W}\} \quad (3)$$

The regression model is trained to learn the mapping function f from the m operational indicators to the n response sensors. For each windowed and scaled input $X_{op}^{(i)}$, the prediction $Y_{pred}^{(i)}$ is given in Eq. (4):

$$Y_{pred}^{(i)} = f(X_{op}^{(i)}) \quad (4)$$

The regressor is trained by minimizing the L1 loss, also referred to as Mean Absolute Error (MAE) due to its robustness to outliers, defined as:

$$Loss_{L1} = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i| \quad (5)$$

The residual window $R^{(i)}$ is then computed as the difference between the actual response sensors window and the predicted one, as shown in Eq. (6):

$$R^{(i)} = Y_{actual}^{(i)} - Y_{pred}^{(i)} \quad (6)$$

These residuals serve as the basis for the anomaly detection autoencoder.

2.2. LSTM-based autoencoder

The residual windows from the regression model, defined in Eq. (6), are fed into an LSTM autoencoder, which is trained using the L1 loss defined in Eq. (5) to reconstruct the residual windows. The encoder E maps a residual window $R^{(i)}$ into a latent space z, and the decoder D reconstructs them:

$$z = E(R^{(i)}), \qquad R_{recon}^{(i)} = D(z) \quad (7)$$

The reconstruction error δ for window i is defined element-wise as:

$$\delta^{(i)} = \left|R^{(i)} - R_{recon}^{(i)}\right| \quad (8)$$

To aggregate overlapping windows back into a continuous time series, we use a weighted averaging approach with a Hann (cosine) window to reduce edge artifacts. The Hann weights $\omega \in \mathbb{R}^W$ defined as:

$$\omega_t = 0.5\left[1 - \cos\left(\frac{2\pi(t-1)}{W-1}\right)\right], t = 1,2,\ldots,W \quad (9)$$

The aggregated reconstruction error δ_t for the original series of length T is computed by a weighted sum over all overlapping windows, normalized by the sum of weights at each time step:

$$\delta_t = \frac{\sum_{i\epsilon I(t)} \omega_{t-(i-1)S} \cdot \delta_{t-(i-1)S}^{(i)}}{\sum_{i\epsilon I(t)} \omega_{t-(i-1)S}}, t = 1,.,T \quad (10)$$

Here, $I(t)$ is the set of all window indices i such that t falls within the i^{th} window $[(i-1)S+1, (i-1)S+W]$. This weighted aggregation ensures smooth transitions between windows and preserves temporal continuity in the reconstructed series.

2.3. Anomaly detection

The anomaly detection threshold $T^{(j)}$ for sensor j is determined from the distribution of reconstruction errors on the validation set, $\delta_{val}^{(j)}$, using the 99th percentile:

$$T^{(j)} = \mathrm{percentile}(\delta_{val}^{(j)}, 99\%) \quad (11)$$

A time step t is labeled as anomalous for sensor j if its reconstruction error exceeds this threshold:

$$Anomaly^{(j)}(t) = \begin{cases} 1, & \delta_t^{(j)} > T^{(j)} \\ 0, & otherwise \end{cases} \quad (12)$$

This method enables per-sensor anomaly detection, combining regression residuals with temporal reconstruction to reliably identify deviations from normal behavior.

2.4. Evaluation Approach

Since the anomaly detection problem is unsupervised and explicit labels are not available, evaluation is performed using a threshold-based strategy (Figure 3). For each response sensor, an anomaly threshold is computed as the 99th percentile of the reconstruction errors on healthy validation data (Eq. (11)). During testing, residual reconstruction errors from the LSTM autoencoder exceeding this threshold are flagged as anomalous (Eq. (12)). Anomaly events are formed by grouping consecutive anomalous points. If the gap anomalous points exceed a predefined minimum, separate events are created. Each event is characterized by its duration (number of points) and its severity (average reconstruction error). This allows short-lived, low-severity events, likely caused by noise or prediction artifacts, to be filtered out, ensuring that detected anomalies correspond to meaningful deviations from normal behavior.

To assess the validity of the detected anomalies, the flagged events are cross-referenced with textual maintenance remarks recorded for each test, which indicate the presence of known abnormal events. While these remarks are not precise labels, they provide qualitative evidence for validating the model's predictions. Furthermore, the sensor exhibiting the highest frequency of detected anomalies is considered the primary indicator of abnormal system behavior. This approach allows

both the identification of anomalies and their probable locations, despite the lack of labeled faulty data.

Algorithm 1: Evaluation Approach with Per-Sensor Threshold

Input: Reconstruction errors $\delta_t^{(j)}$ for each sensor j at time t, test remarks R
Output: Identified sensor responsible for anomaly detection
for *each sensor* j **do**
  Compute threshold $T_j \leftarrow 99^{\text{th}}$ percentile of validation errors $\delta_{val}^{(j)}$;
  Flag an anomaly at time t if $\delta_t^{(j)} > T_j$;
  Merge consecutive anomalies into events;
  Remove minor events (short duration or low severity);
  Count anomaly events C_j for sensor j;
$j^* \leftarrow \arg\max_j C_j$; `// Sensor with the most anomaly events`
if *anomaly periods of* j^* *overlap with remarks* R **then**
  Confirm j^* as primary anomaly detector;
else
  No confirmed match between detected anomalies and remarks;

Figure 3. Evaluation approach algorithm

3. EXPERIMENT SETUP

This section describes the dataset specifications, the inherent challenges, and the sensors installed in the jet engine.

3.1. Dataset Description

The dataset used in this study was collected from a PTE-1200A2 gas turbine engine, which is capable of producing a maximum thrust of 200kgf and is equipped with 16 injectors. The dataset encompasses five operational phases, each representing a different configuration of replaced and non-replaced components during scheduled maintenance cycles shown in Table 1. These phases capture variations in the mechanical condition of the engine, enabling the analysis of fault indicators. Unlike commonly used public datasets such as C-MAPSS (Saxena & Goebel, 2008) or simulated turbofan data, which are generated from high-fidelity simulations or scaled-down models , our dataset captures measurements from an actual engine operating in a controlled laboratory environment. This distinction is critical because real engine behavior includes nuanced transient effects, sensor noise, and interactions between components that are not fully represented in simulations.

Table 1. Operational phases and non-replaced components.

Phase	Non-replaced Components
1	None (clean baseline)
2	Turbine rotor, casing, shaft
3	Inlet, compressor impeller, diffuser casing, shaft
4	Inlet, combustion chamber, evaporation tube, oil separation ring
5	Inlet, bearing, casing

For the purposes of this work, Phase 5 was excluded due to the absence of experiments with sufficiently long operating durations.

Within each of the remaining phases, multiple experiments were conducted under one of two predefined throttle profiles, referred to as Spectrum A and Spectrum B shown in Figure 4. Each experiment lasted approximately one hour, providing continuous time-series sensor measurements for parameters such as vibration, temperature, pressure, and rotational speed (RPM), sampled at approximately 25 measurements per second. These recordings capture both steady-state and transient operating conditions, providing a realistic representation of engine behavior under varying configurations and loads.

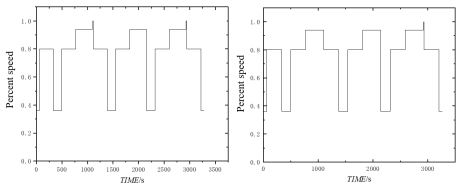

Figure 4. Spectrum A and Spectrum B throttle settings.

3.2. Dataset Challenges

The dataset presents several challenges for anomaly detection in the PTE-1200A2 engine experiments. First, there is a significant class imbalance, as faulty or anomalous events are rare compared to normal operation, making supervised learning approaches difficult. Second, sensor readings contain noise and occasional small discrepancies, which can obscure subtle anomalies. Third, variability in operating parameters across different experiments adds complexity, as it becomes difficult to distinguish whether observed changes are due to normal operational variations or actual anomalies. Finally, the dataset lacks labeled faulty data, with only textual remarks available for some experiments. This absence of

Commented [MR3]: This is very important, mention it after highlight the issue of how we got the ground truth labels
Commented [NM4R3]: Addressed
Commented [MR5]: This section need to be moved after proposed approach
Commented [NM6R5]: Addressed

precise labels necessitates the use of fully unsupervised methods for detecting anomalous behavior.

3.3. Sensors

The engine dataset includes multiple sensors measuring key parameters at various locations to capture the engine's operational state, forming a multivariate time series dataset. Table 2 summarizes the sensor types, measurement ranges, accuracy, and likely locations.

Some of the sensors are considered operational indicator sensors because they primarily reflect the engine's operating conditions. The remaining sensors are response sensors, which vary according to changes in the operational indicators. In this dataset, the response sensors include Tt9, Tt7, Pt3, Pt7, Tt3, and vibration, while all other sensors are classified as operational indicators.

Table 2. PTE-1200A2 engine sensor summary.

Param	Pts	Type	Range	Acc	Location
n	1×1	Hall	0–300k rpm	±100 rpm	Rotor shaft (Comp/Turb)
Fn	1×1	Tension/ Comp	-10–200 kgf	±0.0 3%	Test stand (thrust)
Wf	1×1	Coriolis	0–5000 g/min	1%	Fuel line to combustor
Tt0	1×1	T-type TC	-40–120 °C	±0.5 °C	Inlet air (free stream)
Hum	1×1	Humidity	0–100% RH	±2% RH	Inlet/free stream
Pt0	1×1	Pressure	0–110 kPa	0.5% FS	Ambient (total inlet)
Vib	1×1	Speed	0–20 mm/s	0.5% FS	Bearing housing
T9	8×1	K-type TC	-40–1150 °C	±0.5 °C	Nozzle outlet (EGT)
Tt7	4×1	K-type TC	-200–1300 °C	±0.5 °C	Nozzle inlet
Pt1	4×3	Press scanner	-10–600 kPa	0.05 % FS	Comp inlet (total)
Pt3	4×1	Press scanner	-10–600 kPa	0.05 % FS	Comp outlet (pre-burner)
Pt7	4×1	Press scanner	-10–600 kPa	0.05 % FS	Nozzle inlet
Ps1	4×1	Press scanner	-10–600 kPa	0.05 % FS	Comp inlet (static)
Tt1	4×3	T-type TC	-200–500 °C	±0.5 °C	Comp inlet
Tt3	4×1	K-type TC	-200–1300 °C	±0.5 °C	Comp outlet

4. RESULTS

In this section three anomalous cases are presented: Oil pump air bubbles, Low-speed vibration, and Diffuser crack. The model predictions are evaluated against the textual remarks. Additionally, in the case of low-speed vibration, the evaluation metrics are presented.

4.1. Oil Pump Air Bubbles

This case corresponds to the occurrence of air bubbles in the oil pump, which prevented the engine from reaching maximum speed. We compare the regression model predictions and autoencoder reconstruction for multiple sensors to evaluate model performance during this anomaly.

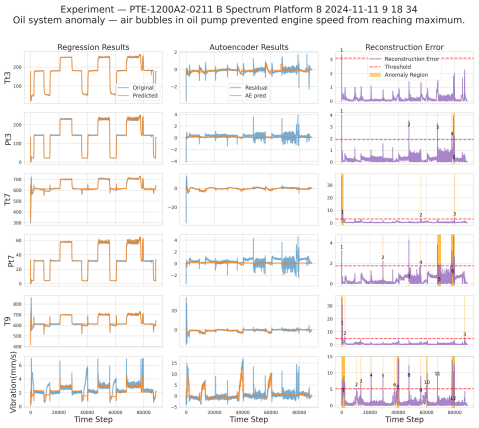

Figure 5. Regression and Autoencoder Results. Left column: Original vs. predicted sensor signals from the regression model. Middle column: Residual signal and autoencoder (AE) predictions, highlighting how the AE reconstructs the residuals. Right column: Reconstruction error with 99th percentile thresholds, where shaded regions indicate detected anomalies.

The Pt7 and Vibration sensors shown in Figure 5 provide clear indicators of the bubbles in the oil pump. The Pt7 sensor measures the engine airflow pressure, and the presence of air bubbles in the pump indirectly affects engine performance, causing small deviations in airflow that are captured as differences between predicted and measured signals by the regression model and as reconstruction errors by the autoencoder. The Vibration sensor detects mechanical vibrations of the pump and engine, which increase and become irregular due to reduced lubrication and cavitation caused by the bubbles. Figure 6 highlights the anomalies in the Pt7 and Vibration sensors. These sensors show anomalous events that are either long-lasting or have high magnitude, whereas the other sensors only have single-step

events or low-severity values. The behavior of these sensors matches the ground truth remark of bubbles in the oil pump, confirming the detected anomaly.

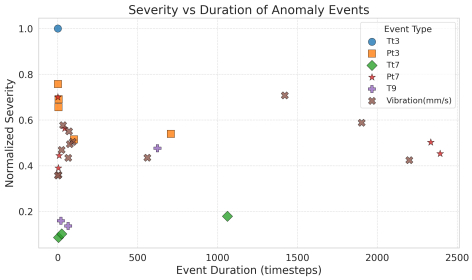

Figure 6. Magnitude of the reconstruction error over the duration of anomaly events for each sensor.

4.2. Diffuser Crack

In this second experiment, the diffuser section is suspected to have a crack. Such a crack can disturb the smooth airflow, creating turbulence and local pressure fluctuations that propagate downstream, which are captured by Pt7 at the nozzle inlet. In addition, the structural disturbance generates mechanical vibrations, observed in the vibration sensor, indicating abnormal loading.

The anomaly detection results in Figure 7 confirms these effects. Slight pressure variations in Pt7, absent in healthy experiments, flag anomalies during sudden rotational speed changes. Similarly, a large portion of the vibration sensor is identified as anomalous, with 15 relatively long events, reflecting the mechanical imbalance caused by the crack. These observations demonstrate that the pipeline successfully detects both aerodynamic and structural anomalies consistent with a diffuser crack. The simultaneous detection in pressure and vibration channels validates that the anomaly detection approach is effective and capable of identifying real faults in the system, providing a reliable early warning tool for diffuser degradation.

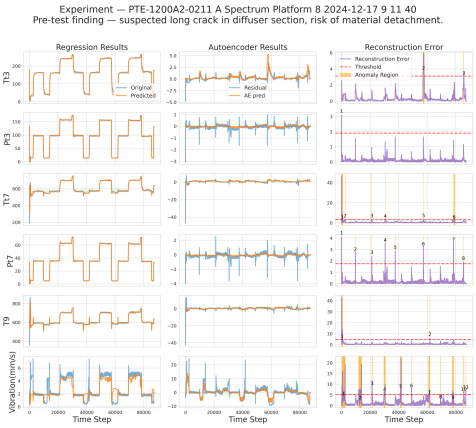

Figure 7. Results of the anomaly detection for the experiment with a crack in the diffuser section.

4.3. Low-Speed Vibration

The third study case corresponds to abnormal vibration within the system occurring at low rotational speeds. Specifically, three Phase 3 experiments were annotated in which the vibration values exceeded the predefined safety limits while the engine's rotational speed remained below 22,000 RPM. This scenario represents a low-speed vibration anomaly, highlighting the system's unusual behavior under conditions where high vibration is not typically expected. Studying such events is critical, as they can indicate developing faults that may compromise engine performance or reliability if left undetected.

A threshold analysis was conducted across the three experiments, and the average performance results are summarized in Table 3. The 99th percentile threshold achieved the best performance and was therefore selected for subsequent evaluation.

Table 3. Classification results on the low-speed vibration cases using RAVEN for different thresholds. Acc. for accuracy, Pre. for precision, Rec. for recall; values are displayed in %.

Threshold	Sample-level classification			
	Acc.	Pre.	Rec.	F1
99	**97.3**	**31.1**	**57.4**	**35.7**
97.5	93.1	19.8	58.9	25.5
95	88.7	14.9	63.0	20.8

Figure 8 presents the results of our anomaly detection framework. The three experiments begin with very low starting rotation speeds, a condition that can amplify the

effects of mechanical imbalance and excite resonance modes, leading to pronounced vibrations in both the system and the sensor readings. These anomalies manifest as distinct high peaks at the beginning of the vibration signals, features largely absent in healthy runs and therefore unlikely to be reproduced by the regression model. This discrepancy results in large residuals that the healthy-trained autoencoder cannot reconstruct, producing high reconstruction errors and flagging the corresponding time steps as anomalous. In experiment A (first row), a slight drop in rotation speed around time step 10,000 caused another vibration spike, aligning with the ground truth note on the second slow-speed step that pushed vibration values beyond acceptable limits. In the C (third row), a final anomaly occurred near the end of the run; this was also a true positive, as the original vibration readings again exceeded the threshold due to excessively low speed, requiring the experiment to be paused and the rotation speed readjusted above 20,000. Across all three experiments, every low-speed vibration anomaly was correctly detected, with no false positives observed.

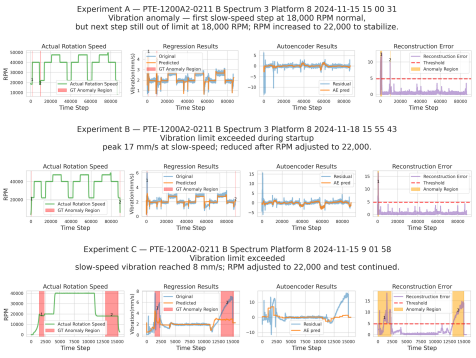

Figure 8. Low-speed vibration results for three annotated experiments (A, B, and C). Column 1: Actual Rotation Speed during the experiment. Column 2: Original vs. predicted vibration signals from the regression model. Column 3: Residual signals and AE reconstructions. Column 4: Reconstruction MAEs with thresholds and anomaly events.

Additionally, these remarks are quantifiable, as they define a vibration limit for slow-speed steps (below 22,000 rpm). To construct the ground truth classes, we set this limit at 4 mm/s and label as anomalous all low-speed time steps exceeding it. Consecutive anomalous time steps are grouped into anomalous events. This ground truth enables us to evaluate the anomaly detection framework in three experiments, both at the sample level and the event-based level. At the sample level, the classification compares the ground truth label with the predicted label for each time step. At the event-based level, an event is considered correctly predicted if any of its points overlap with a ground truth event, allowing for a small temporal window on each side. This matching strategy yields the classification metrics shown in Table 4.

Table 4. Classification results on the low-speed vibration cases using RAVEN. Exp. stands for experiment, Acc. for accuracy, Pre. for precision, Rec. for recall; values are displayed in %.

Exp.	Sample-level classification				Event-based classification			
	Acc.	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1
A	98.3	0.0	0.0	0.0	100	100	100	100
B	99.5	12.8	82.6	22.1	50.0	100	50.0	66.7
C	94.0	80.7	89.7	85.0	100	100	100	100
Avg.	**97.3**	**31.1**	**57.4**	**35.7**	**83.3**	**100**	**83.3**	**88.9**

At the sample level, accuracy is high due to the severe class imbalance between anomalous and healthy samples. However, precision, recall, and F1-score remain low, as detecting short anomalies at the exact timestep is challenging. At the event level, this limitation is mitigated by associating nearby events, resulting in the detection of all anomaly events except one short occurrence at the end of experiment B.

4.4. Performance comparison

In this subsection, the data from the low-speed vibration experiments are used to evaluate the performance of our proposed model against baseline models. RAVEN is compared against simple but widely used unsupervised baselines that represent both statistical and neural approaches One-Class SVM, a basic autoencoder, and an LSTM-based variational autoencoder (LSTMVAE). For the One-Class SVM, the model was trained on 1 million randomly selected samples, with each sample representing a single timestep containing 15 features. For evaluation, the anomaly status of each row (timestep) in the three test files was predicted, anomalous predictions were grouped into events using a gap threshold of 1000, and performance was assessed at both the point (sample) level and event level using a window acceptance of 500. The results of the OneClassSVM are presented in Table 5.

Table 5. Classification results on the low-speed vibration cases using OneClassSVM. Exp. stands for experiment, Acc. for accuracy, Pre. for precision, Rec. for recall; values are displayed in %.

Exp.	Sample-level classification				Event-based classification			
	Acc.	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1
A	99.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0
B	99.6	10.6	63.8	18.1	100	100	100	100
C	85.0	59.0	68.3	63.3	66.7	66.7	100	80.0
Avg.	**94.8**	**23.2**	**44.0**	**27.1**	**55.6**	**55.6**	**66.7**	**60.0**

Commented [MR7]: The experiments need to be explained clearly, second you need baseline comparison you need subsection as comparison against baelines: ML based baselines, and other basic autoencoder approaches and then your method with the name of your method reflectred

Commented [NM8R7]: Experiment is explained. Comparison models remain.

Commented [NM9R7]: Addressed

For the basic autoencoder, the healthy samples were split into training and validation sets. A fully connected autoencoder was implemented with an encoder architecture of [15 → 32 (ReLU) → 16 (ReLU) → 8], mirrored in reverse for the decoder. After training, a reconstruction error threshold was computed at the 99th percentile of the healthy validation data, and the same anomaly detection and evaluation process was applied to the test files. The results of the basic autoencoder are shown in Table 6.

Table 6. Classification results on the low-speed vibration cases using BasicAE. Exp. stands for experiment, Acc. for accuracy, Pre. for precision, Rec. for recall; values are displayed in %.

Exp.	Sample-level classification				Event-based classification			
	Acc.	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1
A	99.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0
B	99.9	0.0	0.0	0.0	50.0	100	50.0	66.7
C	80.6	15.2	0.6	1.1	25.0	33.3	50.0	40.0
Avg.	**93.4**	**5.1**	**0.2**	**0.4**	**25.0**	**44.4**	**33.3**	**35.6**

For LSTMVAE, The dataset is split into training, validation, and testing sets with an 80-10-10 ratio. The training set is used to train both the sequence regressor and the LSTMVAE model, the validation set is used for hyperparameter tuning and threshold selection, and the testing set evaluates performance on unseen anomalies. The LSTMVAE consists of a two-layer LSTM encoder and decoder, each with 64 hidden units and a dropout of 0.2, which processes 500-timestep residual sequences and learns a 32-dimensional latent representation. The model is trained using a beta-VAE loss that combines mean squared error for reconstruction with a KL divergence term (weighted by β=0.5) to regularize the latent space, enabling robust anomaly detection based on reconstruction error. The results are shown in Table 7.

Table 7. Classification results on the low-speed vibration cases using LSTMVAE. Exp. stands for experiment, Acc. for accuracy, Pre. for precision, Rec. for recall; values are displayed in %.

Exp.	Sample-level classification				Event-based classification			
	Acc.	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1
A	95.9	0.1	4.6	0.3	28.6	28.6	100	44.4
B	88.6	0.6	100	1.3	22.2	22.2	100	36.4
C	79.5	47.9	99.9	64.8	66.7	66.7	100	80.0
Avg.	**93.4**	**5.1**	**0.2**	**0.4**	**25.0**	**44.4**	**33.3**	**35.6**

Finally, the performance of RAVEN is compared with both baselines across all evaluation metrics. This comparison highlights the relative strengths of each method and demonstrates the advantages of RAVEN under varying conditions. The consolidated results are summarized in Table 8.

Table 8. Comparison of classification results on the low-speed vibration cases. Exp. stands for experiment, Acc. for accuracy, Pre. for precision, Rec. for recall; values are displayed in %.

Method	Sample-level classification				Event-based classification			
	Acc.	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1
RAVEN	97.3	31.1	57.4	35.7	83.3	100	83.3	88.9
OCSVM	94.8	23.2	44.0	27.1	55.6	55.6	66.7	60.0
BasicAE	93.4	5.1	0.2	0.4	25.0	44.4	33.3	35.6
LSTMVAE	88.0	16.3	68.17	22.1	39.2	39.2	100	53.6

Overall, the comparison demonstrates that RAVEN achieves superior performance relative to all baseline models. At the sample level, RAVEN attains the highest accuracy (97.3%) and F1-score (35.7%), showing a stronger precision–recall balance compared to One-Class SVM, the basic autoencoder, and the LSTMVAE. While the LSTMVAE achieves relatively high recall (68.2%), its precision remains lower, leading to a modest F1-score. At the event level, RAVEN further distinguishes itself with perfect precision (100%) and the highest F1-score (88.9%), highlighting its reliability in detecting true anomalies without false positives. In contrast, the basic autoencoder and One-Class SVM show limited detection capability, and LSTMVAE, despite achieving full recall, suffers from low precision. These results confirm that RAVEN provides a more robust and consistent framework for anomaly detection under variable engine conditions.

5. DISCUSSION

This section discusses the dataset limitations alongside the strengths of the proposed approach, including event-level analysis, sensor-level analysis, and the trade-offs associated with false positive detection.

5.1. Strengths of the Proposed Framework

Despite these limitations, the proposed framework successfully detects anomaly events that align with domain knowledge of the physical system. This allows detected anomalies to be interpreted and analyzed to determine their likely root causes, as demonstrated in the results section. Compared to alternative approaches, the combination of regression-based prediction with residual learning via an autoencoder offers advantages in capturing both temporal dependencies and complex non-linear system behavior. This hybrid approach enhances sensitivity to deviations that traditional threshold-based or purely statistical models might overlook.

5.2. Event-Level Analysis

To better understand the detection behaviour, anomaly events were categorized into three types:

- **Very short-duration events** (< 1 s): These may represent genuine anomalies but are also more

susceptible to prediction artifacts, sensor glitches, or transient noise, increasing the likelihood of false positives.

- **Non-severe anomalies** (low average severity): These events may be caused by measurement noise or subtle fluctuations. In the first case, false positives can arise due to values being close to the detection threshold. In the second case, the events represent true anomalies that are inherently difficult to detect, again due to their proximity to the threshold.
- **Long and severe anomalies** (high magnitude, extended duration): These are the most likely true positives, as sustained deviations of significant magnitude typically correspond to genuine faults or abnormal operating conditions.

By classifying events in this manner, it becomes possible to prioritize which anomalies warrant further investigation or maintenance action. Long and severe events can be addressed with higher urgency, while short or low-severity events may require contextual validation before intervention.

5.3. Sensor-Level Analysis

At the sensor level, analysis reveals varying degrees of importance for anomaly detection. Some sensors are highly critical, capturing high-severity, long-duration anomalies and thus providing the most valuable signals for fault identification. Others are moderately informative, often registering many non-severe events that may reflect noise but can still detect subtle deviations from normal operation. In contrast, some sensors are less reliable, displaying unstable readings with short-duration events that increase susceptibility to false positives. This variation in reliability and informativeness can be effectively visualized using a heatmap or scatterplot, as exemplified in Figure 8, which maps sensors against event severity and duration to highlight their relative contribution to detection performance.

5.4. False Positive Trade-offs in Safety-Critical Applications

While our unsupervised approach may generate a higher number of false positives due to the absence of labeled training data, this limitation is acceptable and even preferable in the context of jet engine monitoring. In safety-critical applications such as aviation, false positives where the system flags normal operation as potentially anomalous are significantly more tolerable than false negatives, where actual developing faults go undetected. A false positive may lead to unnecessary but precautionary maintenance inspections, whereas a false negative could result in catastrophic engine failure with severe consequences for flight safety. Therefore, our model's tendency toward conservative anomaly detection aligns with the fundamental principle that it is better to err on the side of caution when human lives and expensive equipment are at stake.

5.5. Limitations

A key limitation of this study is the limited availability of ground truth anomaly labels, which prevents quantitative evaluation of detection accuracy. Additionally, it is possible that some anomalies were present in the files labeled as healthy; in such cases, the model may have inadvertently learned certain abnormal behaviors. This can reduce recall, as some true anomaly events remain undetected (false negatives). False positives may also occur, arising from sensor noise, transient fluctuations, or the model reacting to minor deviations near the detection threshold that do not correspond to genuine system faults. Furthermore, the choice of model assumptions and the threshold selection process can influence detection sensitivity. The use of more adaptive or dynamically tuned thresholds could potentially improve performance.

Commented [MR10]: I would suggest moving limitations to be just before conclusion or shorten it and merge it with conclusion

Commented [NM11R10]: Addressed

6. CONCLUSION

This study presents a residual-based anomaly detection framework (RAVEN) for early-stage jet engine fault detection under real-world conditions with scarce faulty data. By combining regression-based prediction with LSTM autoencoder residual learning, the method detects deviations without requiring fault examples and addresses challenges such as sensor noise, operational variability, and the absence of ground truth labels. Event-level classification further distinguishes short/non-severe from long/severe anomalies, supporting prioritization of maintenance actions, while sensor-level analysis identifies the most informative measurements for fault detection. RAVEN achieves 97.3% accuracy and an F1-score of 35.7% at the sample level, and 83.3% accuracy, 100% precision, 83.3% recall, and an F1-score of 88.9% at the event level. Beyond accuracy and robustness, RAVEN shows potential for practical deployment in health and usage monitoring systems (HUMS). Its ability to provide event-level alerts allows integration into maintenance workflows as an early-warning system, where detected anomalies can automatically flag components for inspection before fault escalation. The sensor-level analysis supports fault localization by highlighting subsystems most strongly associated with anomalies, guiding targeted diagnostics. Furthermore, the framework operates on lightweight recurrent architectures with manageable computational requirements, enabling near real-time inference suitable for on-board or ground-based monitoring. Future work should involve collaboration with domain experts to refine alert thresholds, validate latency under operational constraints, and extend subsystem-level fault classification, thereby advancing unsupervised anomaly detection toward practical, deployable solutions for complex propulsion systems.

NOMENCLATURE

X_{op}	operational indicator sensors
m	number of operational sensors
T	number of time steps
$min(x^{(j)})$	minimum value of sensor j
$max(x^{(j)})$	maximum value of sensor j
W	window length
S	window stride
$X_{op}^{(i)}$	window input
i	window number
m	operational sensors
W	consecutive time steps
f	mapping function
m	operational indicators
n	response sensors
W	consecutive time steps
$Y_{pred}^{(i)}$	prediction
$L1$	Loss
$R^{(i)}$	residual window
$Y_{actual}^{(i)}$	actual response sensors window
$Y_{pred}^{(i)}$	predicted response sensors window
E	encoder
$R^{(i)}$	residual window
z	latent space
D	decoder
δ	reconstruction error
$I(t)$	set of all window indices
$T^{(j)}$	anomaly detection threshold
j	sensor name
$\delta_{val}^{(j)}$	distribution of reconstruction errors
n	hall sensor
Fn	tension/comp sensor
Wf	coriolis sensor
Tt0	T-type sensor
Hum	humidity sensor
Pt0	pressure sensor
Vib	speed sensor
T9	K-type sensor
Tt7	K-type sensor
Pt1	press scanner
Pt3	press scanner
Pt7	press scanner
Ps1	press scanner
Tt1	T-type sensor
Tt3	K-type sensor

REFERENCES

Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. Journal of Network and Computer Applications, 60, 19–31.

Akcay, S., Atapour-Abarghouei, A., & Breckon, T. P. (2018). GANomaly: Semi-supervised anomaly detection via adversarial training. In Asian Conference on Computer Vision (pp. 622–637). Springer.

Bahri, M., Salutari, F., Putina, A., & Sozio, M. (2022). AutoML: State of the art with a focus on anomaly detection, challenges, and research directions. International Journal of Data Science and Analytics, 14(2), 113–126.

Box, G. (2013). Box and Jenkins: Time series analysis, forecasting and control. In A Very British Affair: Six Britons and the Development of Time Series Analysis During the 20th Century (pp. 161–215). Springer.

Davari, N., Veloso, B., Ribeiro, R. P., Pereira, P. M., & Gama, J. (2021). Predictive maintenance based on anomaly detection using deep learning for air production unit in the railway industry. In 2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA) (pp. 1–10). IEEE.

Du, M., Li, F., Zheng, G., & Srikumar, V. (2017). DeepLog: Anomaly detection and diagnosis from system logs through deep learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (pp. 1285–1298). ACM.

Fernando, T., Gammulle, H., Denman, S., Sridharan, S., & Fookes, C. (2021). Deep learning for medical anomaly detection–a survey. ACM Computing Surveys (CSUR), 54(7), 1–37.

Görnitz, N., Kloft, M., Rieck, K., & Brefeld, U. (2013). Toward supervised anomaly detection. Journal of Artificial Intelligence Research, 46, 235–262.

Hill, D. J., & Minsker, B. S. (2010). Anomaly detection in streaming environmental sensor data: A data-driven modeling approach. Environmental Modelling & Software, 25(9), 1014–1022.

Hodge, V., & Austin, J. (2004). A survey of outlier detection methodologies. Artificial Intelligence Review, 22(2), 85–126.

Huang, H., Wang, P., Pei, J., Wang, J., Alexanian, S., & Niyato, D. (2025). Deep learning advancements in anomaly detection: A comprehensive survey. arXiv preprint arXiv:2503.13195.

Kieu, T., Yang, B., Guo, C., Cirstea, R. G., Zhao, Y., Song, Y., & Jensen, C. S. (2022, May). Anomaly detection in time series with robust variational quasi-recurrent autoencoders. In 2022 IEEE 38th International Conference on Data Engineering (ICDE) (pp. 1342–1354). IEEE.

Kucuk, M. F., & Uysal, I. (2022). Anomaly detection in self-organizing networks: Conventional versus contemporary machine learning. *IEEE Access, 10*, 61744–61752.

Kurz, R., Brun, K., & Wollie, M. (2008). Degradation effects on industrial gas turbines. In Turbo Expo: Power for Land, Sea, and Air (Vol. 43178, pp. 493–504).

Malhotra, P. (2016). LSTM-based encoder-decoder for multi-sensor anomaly detection. CoRR, 1607.

Meng, W., Liu, Y., Zhu, Y., Zhang, S., Pei, D., Liu, Y., Chen, Y., Zhang, R., Tao, S., Sun, P., et al. (2019). LogAnomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs. In IJCAI, 19(7), 4739–4745.

Miao, D., Feng, K., Xiao, Y., Li, Z., & Gao, J. (2024). Gas turbine anomaly detection under time-varying operation conditions based on spectra alignment and self-adaptive normalization. Sensors, 24(3), 941.

Park, D., Hoshi, Y., & Kemp, C. C. (2018). A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder. IEEE Robotics and Automation Letters, 3(3), 1544–1551.

Saxena, A., & Goebel, K. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation (NASA Technical Report No. NASA/TM-2008-215961). NASA Ames Research Center.

Talebi, S. S., Madadi, A., & Tousi, A. M. (2025). Fault diagnosis of micro gas turbine using a hybrid scheme of thermodynamic analysis and artificial neural network. Heliyon, 11(10), eXXXXX.

Tang, S., Yuan, S., & Zhu, Y. (2019). Deep learning-based intelligent fault diagnosis methods toward rotating machinery. IEEE Access, 8, 9335–9346.

Wang, X., & Tong, L. (2022). Innovations autoencoder and its application in one-class anomalous sequence detection. Journal of Machine Learning Research, 23(49), 1–27.

Wei, Y., Jang-Jaccard, J., Xu, W., Sabrina, F., Camtepe, S., & Boulic, M. (2023). LSTM-autoencoder-based anomaly detection for indoor air quality time-series data. IEEE Sensors Journal, 23(4), 3787–3800.

Wong, W.-K., Leckie, C., & Ramamohanarao, K. (2002). Rule-based anomaly pattern detection for detecting disease outbreaks. In *Proceedings of the AAAI/IAAI Conference*.

Zamanzadeh Darban, Z., Webb, G. I., Pan, S., Aggarwal, C., & Salehi, M. (2024). Deep learning for time series anomaly detection: A survey. ACM Computing Surveys, 57(1), 1–42.

Zhou, X., Hu, Y., Liang, W., Ma, J., & Jin, Q. (2020). Variational LSTM enhanced anomaly detection for industrial big data. IEEE Transactions on Industrial Informatics, 17(5), 3469–3477.

BIOGRAPHIES

Nouf Almesafri was born in the United Arab Emirates. She earned her Bachelor of Science in Aerospace Engineering from Khalifa University, Abu Dhabi, in 2022, and her Master of Science in Applied Artificial Intelligence from Cranfield University, United Kingdom, in 2024. Nouf has three years of experience at the Propulsion and Space Research Center, Technology Innovation Institute, Abu Dhabi, where she currently serves as a Senior Associate Researcher. Her research focuses on computer vision, deep learning, and reinforcement learning, with an emphasis on applying advanced AI techniques to aerospace and propulsion systems.

Mohamed Ragab received the Ph.D. degree in Computer Science and Engineering from Nanyang Technological University (NTU), Singapore, in 2022, with a focus on developing robust and transferable AI for real-world predictive maintenance. He is currently a Researcher with the Propulsion and Space Research Center, Technology Innovation Institute (TII), Abu Dhabi, United Arab Emirates, and an Adjunct Researcher with the Agency for Science, Technology, and Research (A*STAR), Singapore. His research interests include industrial AI, transfer learning, machine fault diagnosis, and prognosis. He has been recognized with several honors, including the Finalist Paper Award at the IEEE International Conference on Prognostics and Health Management and the prestigious A*STAR Career Development Award.

Salama Almheiri was born in the United Arab Emirates. She earned her Master of Science in Electrical Power Engineering with distinction from the University of Edinburgh, United Kingdom, in 2021. Salama has two years of experience at the Propulsion and Space Research Center, Technology Innovation Institute, Abu Dhabi, where she currently serves as a Senior Associate Researcher in the Prognostics and Health Management team. Her research focuses on predictive maintenance, edge AI, with applications to aerospace propulsion systems, hydrogen fuel-cell UAVs, and solar powered UAVs.

Dr. Zahi M. Omer is a Senior Researcher at the Propulsion & Space Research Center, Technology Innovation Institute, Abu Dhabi. His work sits at the intersection of applied AI, intelligent control, and predictive analytics for energy and propulsion systems. He develops data-driven and model-informed methods for prognostics and health management, including Remaining Useful Life estimation, and leads hardware-in-the-loop development and real-time validation for photovoltaics, rechargeable battery systems, and smart microgrids. His portfolio spans AI-enabled controllers, edge inference pipelines, and reliability-centric analytics. Dr. Zahi holds a Ph.D. in Electrical Engineering from UAE University and has 12 years of combined industrial and academic experience.

Dr. Abdulla Alseiari is Director of AI Diagnostics & Prognostics at the Propulsion and Space Research Centre, Technology Innovation Institute (TII), Abu Dhabi. He has over 20 years of experience in predictive maintenance and AI applications across the energy and aerospace sectors, including 18 years with TAQA. His research focuses on Prognostics and Health Management (PHM), digital twins, anomaly detection, and decision-support systems for airbreathing propulsion. Dr. Abdulla is a board member at the International Society for Air Breathing Engines (ISABE) and holds a Ph.D. in Maintenance Engineering & AI from the University of Greater Manchester, an MSc in Maintenance Engineering from Glasgow Caledonian University, and a BSc in Electrical Engineering from Al Ain University.