

---

# EARLY FAULT DETECTION ON C-MAPSS WITH UNSUPERVISED LSTM AUTOENCODERS

---

## ABSTRACT

This paper introduces an unsupervised health-monitoring framework for turbofan engines that does not require run-to-failure labels. First, operating-condition effects in NASA C-MAPSS sensor streams are removed via regression normalisation; a Long Short-Term Memory autoencoder is then trained only on the healthy portion of each trajectory. Persistent reconstruction error, gauged with an adaptive threshold derived from training data, triggers real-time alerts without hand-tuned rules. Benchmark results show high recall and low false-alarm rates across multiple operating regimes, demonstrating that the method can be deployed quickly, scale to diverse fleets, and complement Remaining Useful Life models with earlier, confidence-weighted warnings.

**Keywords** aircraft maintenance · predictive maintenance · anomaly detection · unsupervised learning

## 1 Introduction

Predictive maintenance has emerged as a critical strategy for optimising the operational lifespan and reliability of complex mechanical systems. Data-driven approaches enable industries to anticipate failures, reduce downtime, and enhance maintenance scheduling through condition-based monitoring. In aerospace applications, conventional prognostic methods primarily focus on supervised learning techniques for Remaining Useful Life (RUL) estimation. These approaches require complete run-to-failure trajectories to train models that predict the exact operational lifespan of components. However, such comprehensive failure data is often unavailable in real-world scenarios due to safety protocols and maintenance interventions that prevent equipment from reaching complete failure states.

The NASA C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) datasets (Saxena & Goebel, 2008) have become the standard benchmark for developing and evaluating prognostic algorithms in aircraft engines. These datasets simulate realistic operational scenarios and degradation patterns across multiple engine units, providing valuable insights into failure progression. While most existing research concentrates on supervised RUL prediction methods (Fan et al., 2024; Sateesh Babu et al., 2016), the practical challenges of obtaining labelled failure data motivate the exploration of alternative approaches.

This article presents an unsupervised learning approach for early fault detection in aircraft engines using an LSTM autoencoder architecture. The proposed system learns normal operational patterns from sensor data during healthy engine conditions, subsequently flagging deviations that indicate potential degradation. Unlike traditional RUL estimation methods, our approach specifically targets the identification of anomalous behaviour corresponding to the final 10% of engine life, providing maintenance teams with actionable alerts while requiring no labelled failure data for training. The methodology leverages the temporal dependencies inherent in engine sensor readings through the LSTM architecture, offering a practical solution for scenarios where complete run-to-failure data is unavailable.

## 2 Anomaly Detection Approach

Anomaly detection offers a complementary paradigm for condition monitoring that addresses the limitations of supervised RUL prediction. This methodology operates on a fundamentally different principle: instead of attempting to estimate remaining operational time, it identifies deviations from normal operating patterns that may indicate emerging faults. The technique is particularly valuable for early fault detection when complete degradation histories are unavailable.

Within the C-MAPSS framework, unsupervised anomaly-detection methods offer several distinct advantages. Modern techniques—such as deep autoencoders, one-class support vector machines (OC-SVM), isolation forests, and density-based clustering—can model healthy engine behaviour using only data from normal operation (Malhotra et al., 2016). Once trained on baseline (healthy) data, these models detect sensor readings that deviate from established patterns, potentially signalling the onset of degradation (Jakubowski et al., 2021). This paradigm is particularly valuable when early detection is more critical than precise RUL estimation and when labelled failure data are scarce (de Pater & Mitici, 2023). Recent studies have demonstrated the feasibility of deep-learning approaches on aerospace datasets—for instance, LSTM autoencoders have successfully identified incipient faults in simulated aero-engine time-series data (Du et al., 2024). Other researchers report promising results with variational autoencoders and clustering methods for early anomaly detection in comparable prognostic problems (Jakubowski et al., 2021). Leveraging such unsupervised techniques thus enables timely fault detection without run-to-failure labels, complementing traditional RUL models in a practical maintenance strategy.

### 3 Case study

#### 3.1 Data description

The CMAPSS dataset includes turbofan engine degradation data produced using NASA C-MAPSS simulation software (Frederick et al., 2007). This software models the physical behaviour of aero engines under different operating conditions. The dataset consists of degradation samples from multivariate time series, and it is divided into separate training and test subsets. The time series data comes from individual engines of the same type, representing a fleet with varied operational conditions and degradation patterns.

The table 1 shows the different datasets available together with the peculiarities of each one. Set FD001 represents the simplest case with a single operating condition and a single failure mode, the degradation of the HPC (High Pressure Compressor) system. On the other hand, set FD004 is the most complex, consisting of six different operating conditions and two failure modes (HPC and fan degradation).

Table 1: Differences between subsets of the CMAPSS use case.

Subset	Operating conditions	Fault modes	Train engines	Test engines
FD001	1	1	100	100
FD002	6	1	260	259
FD003	1	2	100	100
FD004	6	2	249	248

Each observation in the CMAPSS dataset consists of a timestamped vector of variables, including the engine unit identifier, the operational cycle, three operational condition settings, and 21 sensor measurements. These sensor readings capture various physical properties of the engine such as temperatures, pressures, and rotational speeds at different internal components.

#### 3.2 Data preparation

Before applying any preprocessing techniques, it is essential to partition the dataset to distinguish between the normal operation phase (healthy state) and the degraded operation phase. In this project, we adopt a simple yet effective strategy: the first 85% of each time series is used as the training set. This decision is based on the structure of the CMAPSS dataset, where the initial portion of each engine’s life corresponds to healthy behaviour, while degradation typically begins in the later stages. Reserving the final 15% of each unit’s timeline allows us to focus model evaluation on the period where faults are likely to emerge.

This partitioning is illustrated in Figure 1. The green-shaded region represents the training set, consisting of data from the normal operation phase. The red-shaded region, on the other hand, denotes the validation set that captures the onset and progression of degradation. The vertical red dashed line marks the 85% cutoff point, separating the two phases clearly. This split includes a 5% buffer before the last 10% of remaining useful life (RUL), ensuring a clean boundary between healthy and degraded conditions.

This separation serves three key purposes. First, it ensures that all models are trained exclusively on data representing normal engine operation, establishing a clean baseline of healthy behaviour. Second, it prevents any degradation patterns from contaminating the training process, which is crucial for effective anomaly detection. Third, this methodology

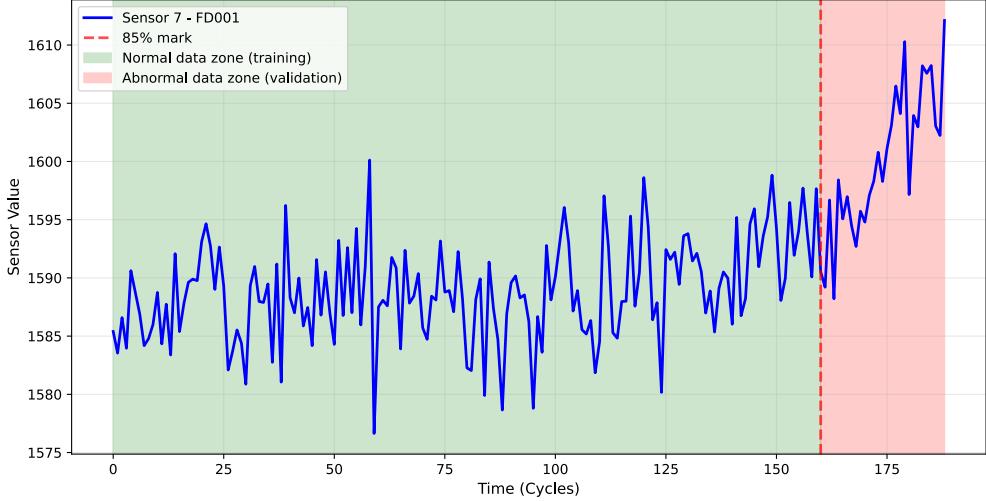


Figure 1: Example of partitioning on sensor #7 set FD001

reflects real-world operational scenarios where systems are typically monitored during normal operation before being deployed to detect emerging faults.

By reserving the final 15% of each time series for validation and testing, we simulate realistic conditions for fault detection. The model’s performance is evaluated on its ability to detect deviations as the system transitions into its failure phase, particularly within the final 10% of the unit’s lifespan—when accurate predictions are most critical. The added 5% buffer before this phase helps reinforce the separation, allowing models to generalise better and reducing the risk of data leakage from early degradation patterns into the training phase.

The CMAPSS datasets FD002 and FD004 present a unique preprocessing challenge due to their inclusion of multiple operational conditions, reflected in three continuous operational variables. These varying conditions significantly influence sensor readings, making direct analysis of raw sensor data inappropriate for degradation modelling. Traditional approaches often cluster operational conditions into discrete states represented by binary indicators. However, we adopt a more sophisticated normalisation approach that explicitly removes operational condition effects while preserving degradation-related patterns.

Our preprocessing methodology follows a regression-based normalisation procedure similar to (Wang et al., 2018, 2019). For each of the 21 sensor signals, we train a Multilayer Perceptron (MLP) regression model that predicts expected sensor values ( $y_{pred}$ ) given the three operational condition variables. The normalisation is then performed by subtracting the condition-dependent predictions from the actual sensor readings:

$$x_{norm} = x_{raw} - y_{pred} \quad (1)$$

where  $x_{raw}$  represents the original sensor measurement and  $x_{norm}$  is the condition-normalised value. This model, trained exclusively on healthy operation data from the training set, learns the conditional relationship between operational parameters and sensor behaviour under normal conditions.

As demonstrated in prior work and visualised in Figure 2, this normalisation reveals hidden trends in sensor data that are otherwise obscured by operational variability. The resulting normalised sensor values better reflect the true health state of each engine, providing cleaner inputs for subsequent autoencoder training and anomaly detection. For FD001 and FD003 datasets (single operational condition), we apply conventional min-max normalisation, as operational condition effects are not present.

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (2)$$

This preprocessing strategy is particularly crucial for our unsupervised approach, as it ensures that the autoencoder learns to reconstruct and detect anomalies in condition-independent degradation patterns rather than operational artefacts. The effectiveness of this method has been validated in previous RUL estimation studies and proves equally valuable for anomaly detection tasks.

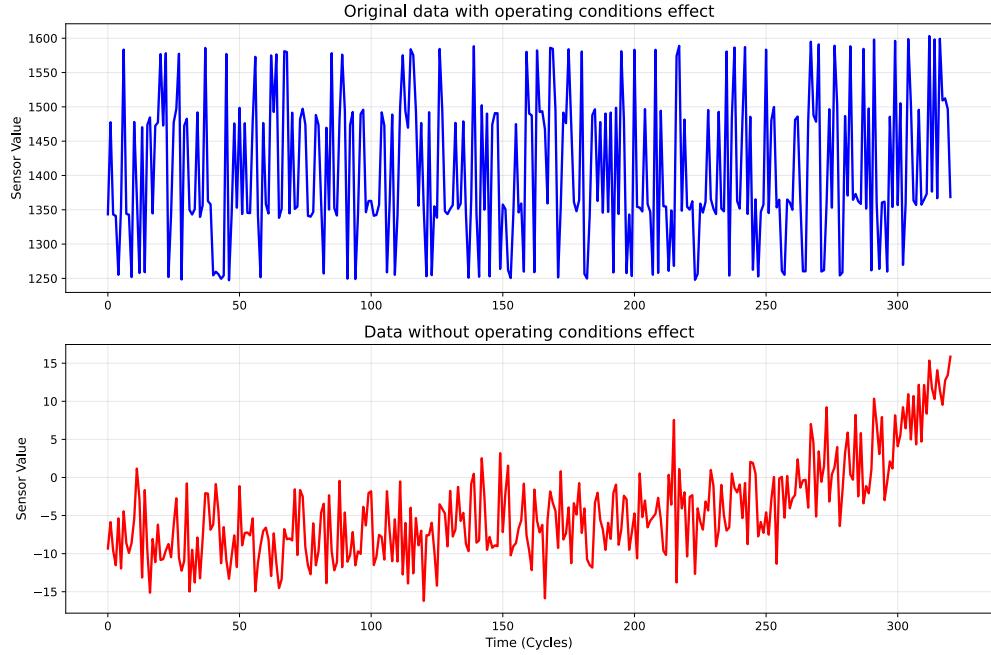


Figure 2: Removing the effect of operating conditions on sensor #7

### 3.3 Exploratory data analysis

The initial exploratory analysis of the CMAPSS dataset reveals several important characteristics of the sensor data. While some sensors show clear temporal patterns related to engine degradation, others appear relatively constant or exhibit minimal variation throughout the operational life of the engines. Traditional supervised approaches might consider eliminating these apparently uninformative sensors to reduce dimensionality and noise. However, our unsupervised approach based on autoencoders takes a fundamentally different perspective.

From the correlation analysis (Figure 3), we observe that several sensor pairs show high collinearity during normal operation, while some sensors demonstrate near-zero variance. Notably, sensors 1, 5, 10, 16, 18 and 19 exhibit particularly low variability across all operational conditions. In a supervised learning context, these would typically be excluded as non-informative features. However, in our unsupervised framework, we intentionally retain all sensors for three key reasons:

First, the autoencoder's latent space representation can automatically learn to weight sensor importance during training, effectively performing implicit feature selection. The network architecture naturally compresses the input dimensions while preserving the most relevant patterns for reconstruction. Second, while some sensors may appear uninformative during normal operation, their behaviour during fault conditions might contain subtle but critical anomalies that would be missed if excluded a priori. Third, the interaction between multiple "low-variance" sensors might collectively contain valuable information that individual sensors lack.

The temporal analysis (Figure 4) shows that most sensors maintain stable baselines during normal operation, with deviations becoming apparent only during the degradation phase. This reinforces our decision to train exclusively on the first 85% of each time series, as it provides the cleanest representation of healthy operation. Interestingly, some sensors that appear constant during normal operation develop clear degradation signatures later in the life cycle, validating our decision to retain all sensors.

The operational condition analysis reveals that sensor behaviours cluster distinctly under different operating regimes, as shown in the original data with operating conditions effect (Figure 2). This finding motivated our operational condition normalisation approach described in Section 3.3, as it confirms that raw sensor values are heavily influenced by external operating parameters rather than solely reflecting engine health state.

This comprehensive EDA supports our methodological decisions to: (1) retain all sensors despite apparent low variability in normal operation, (2) employ an autoencoder architecture capable of learning meaningful latent representations, and (3) implement operational condition normalisation before anomaly detection. The analysis confirms that our

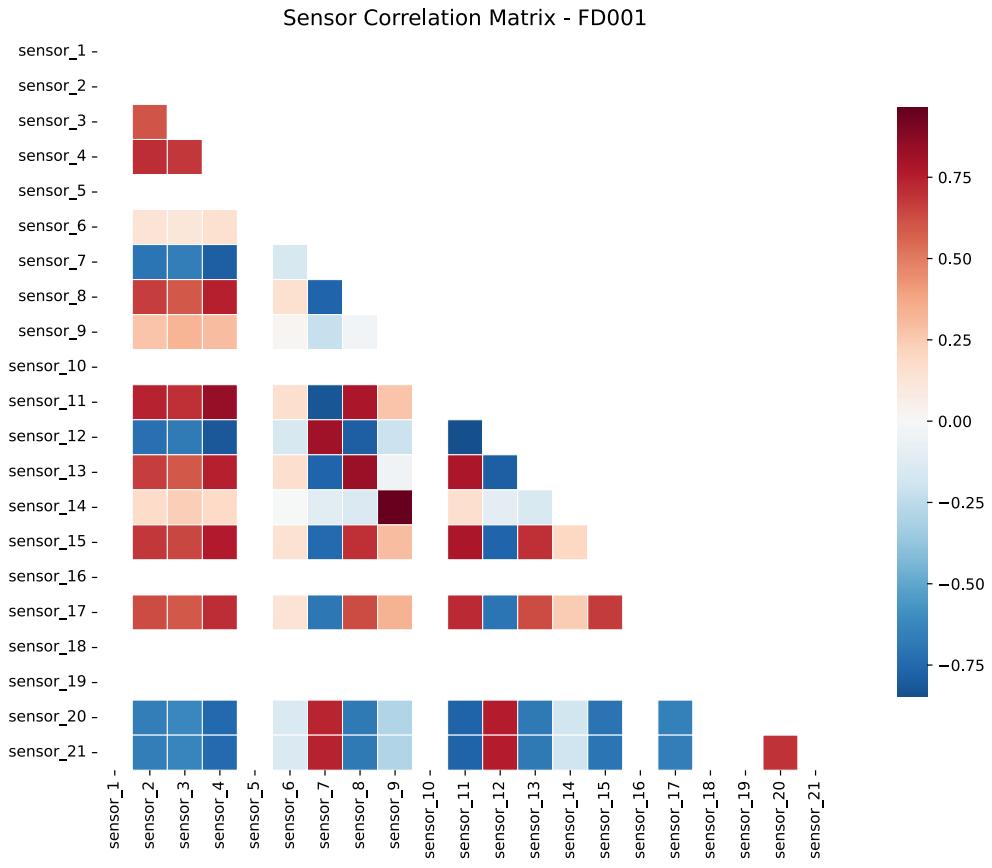


Figure 3: Correlation matrix of sensor readings during normal operation phase

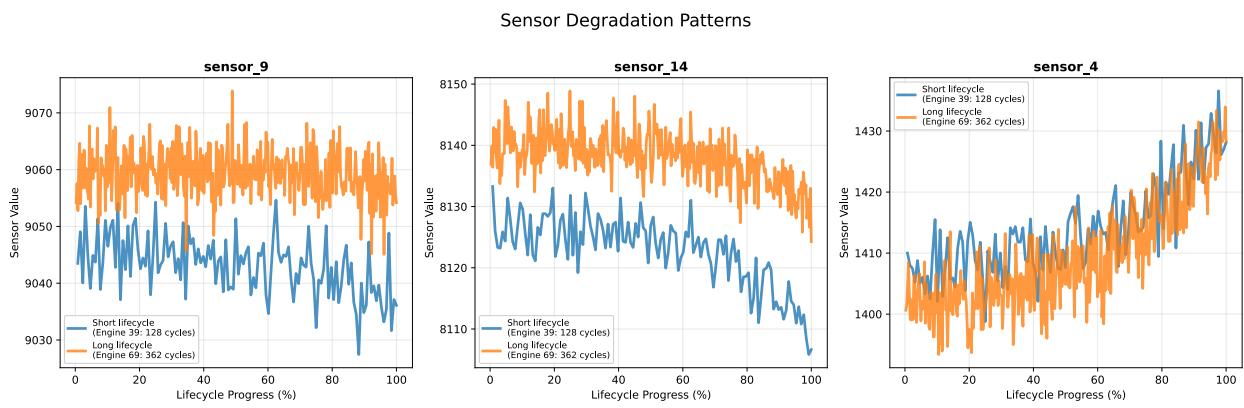


Figure 4: Temporal evolution of selected sensor readings showing normal operation and degradation phases

unsupervised approach can potentially capture subtle degradation patterns that traditional feature selection methods might overlook during manual preprocessing (Jakubowski et al., 2021).

### 3.4 Feature Engineering

Customary feature engineering in predictive maintenance commonly depends on selecting manually and extracting handcrafted features using raw sensor data like statistical moments (mean, variance), time-domain characteristics, or frequency-domain decompositions. However, fault detection's most unique patterns might not be captured always through these methods, especially in aircraft engines and complex systems (Ramasso & Saxena, 2014).

In this work, we do adopt an alternative model. The model in itself performs feature extraction automatically during the training. In particular, we make use of the latent space representation. When trained, a deep LSTM autoencoder neural network learns this representation to reconstruct its input data through a compressed bottleneck layer. For preservation of the most salient features for reconstruction, the encoder component learns to project high-dimensional sensor readings into a lower-dimensional latent space. Nonlinear relationships also temporal dependencies in the data are intrinsically captured by this latent representation without explicit feature engineering.

## 3.5 Model

### 3.5.1 Model Architecture

The proposed anomaly detection system employs a Long Short-Term Memory (LSTM) autoencoder, a specialized neural network architecture particularly suited for capturing temporal dependencies in multivariate time series data. The model consists of two fundamental components:

- **Encoder:** A stack of LSTM layers that processes an input sequence of sensor readings and compresses it into a fixed-length latent vector. At each time step  $t$ , each LSTM cell updates its hidden state  $h_t$  and cell state  $C_t$  using gated operations (forget, input, and output gates) on the current input  $x_t$  and the previous hidden state  $h_{t-1}$ . The final hidden state at the end of the sequence becomes the latent representation (the bottleneck feature vector), capturing the information of the entire window of input data.
- **Decoder:** A corresponding stack of LSTM layers that takes the latent vector and attempts to reconstruct the original sequence of sensor readings. This decoder essentially learns to generate the expected sensor values at each time step given the encoded summary of the sequence.

For the C-MAPSS case study, we implemented a 3-layer LSTM encoder with hidden state dimensions of 16, 8, and 4, respectively. The last encoder layer's output is a 4-dimensional latent vector (bottleneck). The decoder is a mirror of the encoder in architecture, using LSTM layers to expand the 4-dimensional code back to the full sequence dimension. The relatively small size of the latent space (4) forces the network to learn a compact representation of normal engine behaviour, which helps it generalise and detect anomalies as reconstruction errors when patterns deviate from this normal manifold. Similar LSTM-based encoder-decoder architectures have been applied successfully to anomaly detection tasks in other domains, demonstrating the ability of a deep compressed model to learn complex temporal patterns of normal operation

### 3.5.2 Problem-Specific Adaptation

The model is specifically adapted for aircraft engine anomaly detection through three key design choices:

1. **Temporal Window Processing:** Input sequences are structured as sliding windows of 10 operational cycles (approximately 5% of average engine life, depending on the specific dataset 1), allowing the model to learn both instantaneous sensor relationships and their temporal evolution.
2. **Condition-Aware Training:** The preprocessed sensor data (after operational condition normalisation) is augmented with the three operational condition variables as auxiliary inputs, enabling the model to implicitly learn condition-specific normal patterns.

### 3.5.3 Training Protocol

The LSTM autoencoder is *trained exclusively* on healthy data (the first 85 % of the cycles for every engine), and the remaining 15 % is held out solely for post-training evaluation of anomaly detection. All windows extracted from the healthy portion of every engine form the training set. The network minimises the mean-squared reconstruction error (MSE) using the Adam optimiser with an initial learning rate of  $1 \times 10^{-3}$ . Twenty percent of the healthy windows are

randomly set aside as an internal validation split. We employ early stopping: training halts if the validation MSE does not improve for 10 consecutive epochs, preventing overfitting to noise in the baseline behaviour.

Throughout training, care was taken to prevent any information from the test (degradation) phase from leaking into the model training. All hyperparameters and threshold selections were made using only training-phase data and validation splits from that training data, mimicking a realistic scenario where one would train on past fleet data and then deploy the model on new engines.

## 4 Evaluation

### 4.1 Anomaly Detection Mechanism

During operation, the trained LSTM auto-encoder continuously ingests sliding windows of raw sensor measurements and outputs a reconstructed counterpart for each window. Let  $\mathbf{x}_t \in \mathbb{R}^{s \times w}$  denote the stack of  $s$  sensor signals sampled over  $w$  time steps at window index  $t$ , and let  $\hat{\mathbf{x}}_t$  be its reconstruction. The discrepancy between the two is quantified by the window-level mean-squared error, which serves directly as the anomaly score.

$$\text{MSE}_t = \frac{1}{s w} \sum_{i=1}^s \sum_{j=1}^w (x_{t,i,j} - \hat{x}_{t,i,j})^2, \quad (3)$$

To convert this score into a binary decision, we calibrate a threshold using the training set exclusively. From the empirical distribution of their reconstruction errors, we compute the mean  $\mu_{\text{MSE}}$  and standard deviation  $\sigma_{\text{MSE}}$ . The decision boundary is then fixed as

$$\tau = \mu_{\text{MSE}} + \lambda \sigma_{\text{MSE}}, \quad (4)$$

where the multiplier  $\lambda$  (e.g.,  $\lambda = 2.5$ ) is a tunable hyper-parameter. During deployment, any sequence whose reconstruction error exceeds  $\tau$  is flagged as anomalous. Adjusting  $\lambda$  allows us to trade off between the false-alarm rate and missed detections, while anchoring the threshold to the observed variability of genuinely healthy data. The Figure 6 shows the distribution of the errors in the different sets of the FD001 data sets.

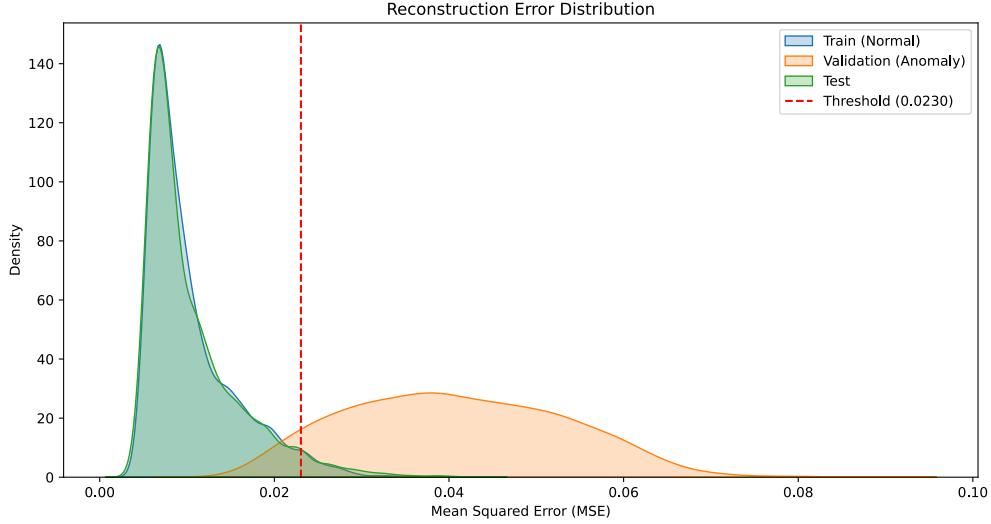


Figure 5: Error distribution - FD001

### 4.2 Evaluation Protocol

Because the original C-MAPSS files do not annotate the exact onset of degradation, we impose an internal partition: the first 90 % of cycles in each trajectory are treated as normal, while the remaining 10 % are labelled degraded. After processing a complete trajectory, the detector's predictions are cross-tabulated against these in-house labels to build the

confusion-matrix counts  $\{TP, FP, TN, FN\}$ . Precision, recall (sensitivity), specificity and  $F_1$ -score are then derived, measuring simultaneously (i) how reliably the system raises an alert during the critical last 10 % of life and (ii) how rarely it emits false alarms during the much longer healthy phase. Together with early-warning-time statistics, these metrics form the quantitative basis of the results discussed in the next section.

## 5 Results

Figure 6 illustrates the end-to-end detection pipeline for a representative engine (unit 81 in subset FD001). The blue curve shows the reconstruction error—here, the mean-squared error (MSE) between the input window and the LSTM-AE reconstruction—computed for each operational cycle. During nominal operation (cycles 1–180) the error fluctuates within a narrow band well below the data-driven anomaly threshold. From cycle  $\sim 185$  onward the error begins to drift upward, crossing the threshold at cycle 195—the point that corresponds to 90 % of the engine’s life (orange dotted line). Every window whose error exceeds the threshold is flagged as an *anomaly* (red markers).

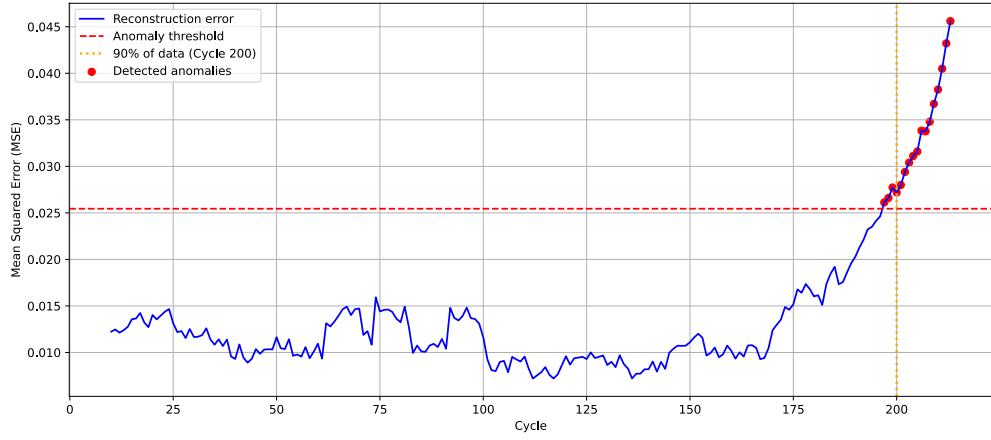


Figure 6: Reconstruction Error per cycle - Engine 81 - FD001

Table 2 summarises the detection performance of the proposed LSTM-AE model on the four C-MAPSS subsets. Although the class imbalance is extreme (between 0.8 % and 1.3 % of the windows in each subset are labelled as anomalous), the auto-encoder achieves very high recall in all cases ( $\geq 0.725$ ) while keeping the specificity above 0.96, indicating a low false-alarm rate during healthy operation.

### Single-regime subsets (FD001, FD003)

For FD001 and FD003—where only one operating regime is present—the model attains perfect or near-perfect recall (0.981 and 1.000, respectively). Precision, however, remains modest (0.196 and 0.241), pulling the  $F_1$ -scores down to 0.327 and 0.388. This behaviour is expected in highly imbalanced settings: a detector tuned for maximum sensitivity will inevitably flag some healthy windows as suspicious, depressing precision (Malhotra et al., 2016). Because false positives are tolerated better than missed failures in safety-critical maintenance, this trade-off is acceptable and comparable to the figures reported in previous unsupervised work on C-MAPSS (de Pater & Mitici, 2023).

### Multi-regime subsets (FD002, FD004)

FD002 and FD004 pose a substantially harder problem: six distinct operating conditions interact with the sensor dynamics, producing a broader range of “normal” patterns. Even after the regression-based normalisation, residual regime effects can shift sensor baselines just enough to mimic fault signatures, complicating the separation between healthy and anomalous windows (Baptista et al., 2021). The model still maintains good specificity ( $\geq 0.982$ ), but recall drops to 0.799 (FD002) and 0.725 (FD004), and the overall  $F_1$ -score falls to 0.510 and 0.456, respectively. Nevertheless, the achieved recall in these multi-regime sets still exceeds the 0.70 benchmark often cited for practical early-warning systems (Yan et al., 2023), while the high specificity ensures that maintenance teams are not overwhelmed by false alarms.

## Discussion of metric interplay

Because precision and recall respond oppositely to the decision threshold, the reported  $F_1$ -scores reflect a conscious bias towards high recall, a common choice in prognostics where missing an incipient fault is costlier than an unnecessary inspection. If required, the threshold could be tightened to boost precision (e.g., for on-wing monitoring with limited maintenance slots), at the expense of detecting some faults slightly later. Finally, we note that the extreme imbalance also deflates the apparent  $F_1$ : despite a score of 0.327 on FD001, the detector correctly flags 98 % of true anomalies while raising false alarms in only 3.4 % of healthy windows.

Table 2: Detection performance on the four C-MAPSS subsets

Dataset	Precision	Recall (Sensitivity)	Specificity	$F_1$ -score	Anomaly %
FD001	0.196	0.981	0.966	0.327	0.8%
FD002	0.374	0.799	0.982	0.510	1.3%
FD003	0.241	1.0	0.963	0.388	1.2%
FD004	0.336	0.725	0.985	0.456	1.1%

## 6 Limitations and Future Work

Although the proposed LSTM-AE delivers high recall with few false alarms, several caveats limit immediate industrial deployment. First, the 85/15 normal-degraded partition is only a heuristic proxy for real operations; engines may transition to fault states at different points, so a principled regime-change detector or Bayesian change-point model is needed to learn boundaries automatically.

Sudden load changes or sensor glitches can produce short-lived false positives. To improve robustness we already propose a *temporal persistence requirement*: an alert is raised only when the reconstruction error exceeds the threshold for at least  $k$  consecutive windows (with  $k = 5, \approx 2\%$  of an engine trajectory). Future work must validate the optimal choice of  $k$  under different duty cycles and cost models.

Even after regression-based normalisation, residual operating-condition effects remain the dominant source of error in the multi-regime datasets FD002 and FD004. Domain-adversarial training or contrastive representation learning could help produce condition-invariant embeddings that generalise better across regimes.

## 7 Conclusion

This work demonstrates that an entirely unsupervised Long Short-Term Memory autoencoder, trained only on baseline data and preceded by a regression-based operating-condition normalisation, can serve as a practical early-fault detector for turbofan engines. By learning the manifold of healthy behaviour and measuring reconstruction error online, the model flags deviations well before traditional Remaining Useful Life (RUL) estimates would approach zero, even though no failure labels are required for training.

Three core contributions underpin this outcome. First, the regression normalisation successfully removes multi-regime artefacts while preserving subtle degradation trends, enabling a single detector to operate across heterogeneous duty cycles. Second, a compact four-dimensional latent representation captures essential engine dynamics, allowing robust anomaly detection with modest computational overhead. Third, the evaluation protocol—training on the first 85 % of life and testing on the final 10 %—closely mirrors industrial practice and shows that the method achieves high recall with a low rate of spurious alerts.

Beyond these technical advances, the framework complements existing RUL models by providing an earlier, confidence-weighted indicator of incipient faults, thereby supporting condition-based maintenance decisions without inflating the false-positive burden. Although further work is needed to adapt thresholds dynamically, enhance interpretability, and bridge the simulator-to-flight gap, the present study offers a label-free foundation on which hybrid, data-plus-physics strategies for safety-critical prognostics can be built.

## References

- Baptista, M. L., Henriques, E. M., & Goebel, K. (2021). A self-organizing map and a normalizing multi-layer perceptron approach to baselining in prognostics under dynamic regimes. *Neurocomputing*, 456, 268–287. <https://doi.org/10.1016/j.neucom.2021.05.031>

- de Pater, I., & Mitici, M. (2023). Developing health indicators and rul prognostics for systems with few failure instances and varying operating conditions using a lstm autoencoder. *Engineering Applications of Artificial Intelligence*, 117, 105582. <https://doi.org/10.1016/j.engappai.2022.105582>
- Du, W., Zhang, J., Meng, G., & Zhang, H. (2024). Aero-engine fault detection with an lstm auto-encoder combined with a self-attention mechanism. *Machines*, 12(12). <https://doi.org/10.3390/machines12120879>
- Fan, Z., Li, W., & Chang, K.-C. (2024). A two-stage attention-based hierarchical transformer for turbofan engine remaining useful life prediction. *Sensors*, 24(3). <https://doi.org/10.3390/s24030824>
- Frederick, D. K., DeCastro, J. A., & Litt, J. S. (2007). *User's guide for the commercial modular aero-propulsion system simulation (cmappss)* (tech. rep.).
- Jakubowski, J., Stanisz, P., Bobek, S., & Nalepa, G. (2021). Anomaly detection in asset degradation process using variational autoencoder and explanations. *Sensors*, 22, 291. <https://doi.org/10.3390/s22010291>
- Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., & Shroff, G. (2016). Lstm-based encoder-decoder for multi-sensor anomaly detection [Proceedings of the ICML 2016 Workshop on Anomaly Detection]. *arXiv preprint arXiv:1607.00148*. <https://arxiv.org/abs/1607.00148>
- Ramasso, E., & Saxena, A. (2014). Review and analysis of algorithmic approaches developed for prognostics on cmappss dataset. *Annual Conference of the Prognostics and Health Management Society 2014*.
- Sateesh Babu, G., Zhao, P., & Li, X.-L. (2016). Deep convolutional neural network based regression approach for estimation of remaining useful life. In S. B. Navathe, W. Wu, S. Shekhar, X. Du, X. S. Wang, & H. Xiong (Eds.), *Database systems for advanced applications* (pp. 214–228). Springer International Publishing.
- Saxena, A., & Goebel, K. (2008). Turbofan engine degradation simulation data set. *NASA ames prognostics data repository*, 18, 878–887.
- Wang, Q., Farahat, A., Ristovski, K., Tang, H.-K., Serita, S., & Gupta, C. (2018). What maintenance is worth the money? a data-driven answer. *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, 284–291. <https://doi.org/10.1109/INDIN.2018.8472003>
- Wang, Q., Zheng, S., Farahat, A., Serita, S., & Gupta, C. (2019). Remaining useful life estimation using functional data analysis. <https://arxiv.org/abs/1904.06442>
- Yan, S., Shao, H., Min, Z., Peng, J., Cai, B., & Liu, B. (2023). Fgdae: A new machinery anomaly detection method towards complex operating conditions. *Reliability Engineering & System Safety*, 236, 109319. <https://doi.org/10.1016/j.ress.2023.109319>