NASA CMAPSS

Engine Health Monitor — GUI User Guide

Complete Notes for Every Tab, Every Chart, Every Feature

Streamlit Web Application | DRDO / GTRE | Turbofan PHM System v2.0

Built with: Python | Streamlit | Matplotlib | Seaborn | XGBoost | Scikit-learn

INFO **What This Document Covers**

This guide explains every tab, every chart, every button, and every number shown in the GUI dashboard.
It is written for DRDO/GTRE engineers who use the system daily -- not for data scientists.
After reading this you will know exactly what to look at, what the colours mean, when to raise an alert, and how to use the Live Sensor Check tool for real-time fault identification.

Section 1 -- GUI Overview: Architecture and How to Start

What the GUI Is

The GUI is a web application that runs entirely on your own computer -- no internet required. It opens in Chrome or Edge like a website, but it is powered by Python running in the background. All data stays local; nothing is sent anywhere.

The application has 6 tabs, each corresponding to one section of the machine learning pipeline. An engineer can use it to monitor the whole engine fleet, inspect individual engine health, explore why the models make their decisions, and test sensor readings in real time.

How to Start the GUI

1 **Run the pipeline first (one-time setup)**

Open a terminal in the project folder and run: python main.py all -- this trains all models, generates predictions, and saves everything. You only need to do this once (or again after new data arrives).

2 **Install Streamlit (one-time, 30 seconds)**
In the same terminal run: pip install streamlit -- this installs the web framework. Only needed once per machine.

3 **Launch the dashboard**
Windows: Double-click launch_gui.bat in the project folder. Linux/Mac: Run bash launch_gui.sh in terminal. The browser opens automatically at http://localhost:8501

4 **Stop the dashboard**
Go back to the terminal window that opened when you double-clicked. Press Ctrl + C to stop the server. The browser page will go offline.

```
# Manual start (if launcher scripts don't work):
$ cd your-project-folder
$ venv\Scripts\activate          # Windows
$ source venv/bin/activate        # Linux / Mac
$ streamlit run app.py

# The terminal will print:
#   Local URL: http://localhost:8501
# Open that URL in Chrome or Edge
```

The Sidebar (Left Panel)

The sidebar is always visible on the left side. It contains three things:

Sidebar Item	What It Shows
Dataset label	Shows which dataset is loaded (FD001, FD002 etc.). Set in config.yaml.
Models: Green/Red	Green means all 4 models are trained and loaded. Red means run python main.py train first.
Data: Green/Red	Green means processed data and predictions exist. Red means run python main.py all first.
Engine selector	Dropdown to pick any engine (001-100). Changing this updates the Dashboard and Live Check tabs instantly.

Section 2 -- Tab 1: Dashboard (Fleet Overview)

HOME Dashboard Tab -- Fleet Overview and Engine Health Cards

Purpose of This Tab

The Dashboard is the first thing you see when the GUI opens. It answers the question: What is the current health status of every engine in the fleet right now? It is designed to be used as a daily morning check -- glance at it to see if any engine needs attention before the day's flights begin.

Part 1: Fleet KPI Cards (Top Row)

The top row shows four numbers covering the entire fleet of engines. These update automatically when you run the pipeline with new data.

KPI Card	What It Means
Total Engines	Total number of engines in the dataset being monitored.
Healthy Engines	Engines where the last predicted RUL is >= 30 cycles. Shows count and percentage of fleet.
Fault-Zone Engines	Engines where predicted RUL < 30 cycles. These are in the danger zone and need immediate attention.
Ensemble Anomalies	Total rows across all engines where BOTH Isolation Forest AND LOF flagged an unusual sensor reading.

Part 2: Selected Engine Health Card

The large card in the centre shows detailed health information for whichever engine you select in the sidebar dropdown. It contains six elements:

Element	What It Means
RUL Countdown (big number)	The predicted number of cycles remaining before this engine needs maintenance. Colour coded: Green >= 30, Amber 10-29, Red < 10.
Status label	HEALTHY (green) / WARNING (amber) / CRITICAL (red) -- shown directly below the RUL number.
Fault Probability %	The XGBoost Fault Classifier's confidence that this engine is currently in the failure danger zone. Above 50% triggers the fault prediction.
Anomaly indicator	Green tick = no anomaly. Red warning = both anomaly detectors agreed this engine shows unusual sensor patterns.

True RUL (if available)	The actual ground-truth RUL from the dataset. Shown alongside the prediction so you can see the error.
Error delta	Predicted RUL minus True RUL. Positive = over-estimated (predicted more life than remains). Negative = under-estimated (safe direction).

Part 3: RUL Timeline Chart

The right side shows two charts stacked vertically for the selected engine:

CHART **RUL Timeline Chart (top)**

Blue solid line = Predicted RUL across all cycles of this engine's life.
Green dashed line = True RUL (actual values from dataset).
Red dotted horizontal line at 30 = The fault threshold. When predicted RUL crosses below this line, fault=1 is predicted.
Amber dotted line at 10 = Critical zone. Engine needs immediate grounding.
What to look for: Do the blue and green lines track closely together? Large gaps = the model is less certain for this engine.

CHART **Fault Probability Chart (bottom)**

Red filled area = Fault probability at every cycle. 0% = definitely healthy, 100% = definitely in fault zone.
Amber dashed line at 0.5 = The decision boundary. Above this = fault predicted.
What to look for: A gradual rise toward the end of engine life is expected and healthy. An early sudden spike is unusual and worth investigating.

Part 4: Fleet Overview Mini-Grid (Bottom)

Scrolling down shows a grid of small cards -- one per engine in the fleet. Each card shows the engine number, its last predicted RUL, and its status label. The border colour of each card matches the status: green border = healthy, amber = warning, red = critical. This lets you scan the entire fleet in seconds and spot any outliers immediately.

GREEN border	AMBER border	RED border
RUL >= 30 cycles. Healthy. No action needed.	RUL 10-29 cycles. Schedule maintenance soon.	RUL < 10 cycles. Immediate attention required.

Section 3 -- Tab 2: M1 Data Analysis

M1 M1 Tab -- Data Loading, Cleaning and Pre-Processing Explorer

Purpose of This Tab

This tab lets you explore the raw data and understand exactly what happened during Milestone 1 pre-processing. You can select any sensor and see its distribution, its trend over an engine's lifetime, and how the RUL labels are distributed across the dataset. It is the foundation that all other milestones build on.

Part 1: Dataset Statistics (Top Row)

Element	What It Does
Total Rows	Number of sensor readings across all engines in the training dataset.
Engines	Number of unique engines in the dataset (100 for FD001).
Active Sensors	Sensors retained after dropping constant ones (14 for FD001 out of 21 original).
Dropped Sensors	Sensors with standard deviation near zero -- they never change so carry zero information (7 dropped).
RUL Cap	Maximum RUL label used during training (125 cycles). Readings beyond this are clipped.

Part 2: Sensor Distribution + Lifecycle Chart

Use the dropdown to select any active sensor (s2, s3, s4, s7 etc.). Two charts appear side by side:

LEFT **Distribution Histogram (left chart)**

Shows how often each sensor value occurs across all 20,631 training rows.
Green dashed line = mean value (average across all readings).
Red dotted lines = 3-sigma bounds (mean +/- 3 standard deviations).
What to look for: A bell-curve shape is good -- it means the sensor has a predictable normal range. Very skewed distributions might indicate a sensor that behaves differently in different operating conditions.

RIGHT **Lifecycle Trend Chart (right chart)**

Shows this sensor's readings across every cycle of the selected engine (chosen in the sidebar).
You can see how the sensor value drifts as the engine ages.
What to look for: Most sensors show a clear gradual trend (rising or falling) as the engine degrades. Flat lines = this sensor doesn't respond to wear. Noisy jumpy lines = the 5-cycle rolling smoothing may need adjustment.

Part 3: Correlation Heatmap

The full-width heatmap below the sensor charts shows how all 14 active sensors relate to each other. Each cell shows the correlation coefficient between two sensors.

Cell Colour	What It Means
Dark red cell (r near +1)	These two sensors move together strongly -- when one rises, the other rises too. They carry overlapping information. This is why PCA (in M3) is used to compress them.
Dark blue cell (r near -1)	These two sensors move in opposite directions. Still correlated (just inversely), still redundant information.
White/pale cell (r near 0)	These two sensors are independent -- they carry different information about the engine.
Diagonal cells (r = 1.0)	A sensor always correlates perfectly with itself. These are always dark red and are expected.

Part 4: RUL Distribution Chart

The bottom chart shows how RUL values are distributed across the entire training dataset. Understanding this distribution explains many of the model's design choices:

DIST **Reading the RUL Distribution**

Red dashed line at 30 cycles = the fault threshold. All rows to the LEFT of this are labelled fault=1.

Amber dashed line at 125 cycles = the RUL cap. Rows beyond this are clipped to 125 because engine sensor readings are essentially identical for RUL=200 vs RUL=500 -- the model cannot distinguish them anyway.

The distribution is flat from 0-125, meaning every RUL value is roughly equally represented in the training data. This is by design -- the clipping creates a uniform distribution.

Section 4 -- Tab 3: M2 Sensor Characterisation

M2 M2 Tab -- Sensor Analysis, T-Tests, and Health Index

Purpose of This Tab

This tab answers the fundamental question: which sensors actually change as the engine wears out? It uses statistical tests to prove that the difference between a new engine and a worn engine is real and not just random noise. It also shows the Health Index -- a single number that summarises overall engine condition.

Part 1: Healthy vs Degraded Comparison Charts

Select any sensor from the dropdown. The system splits every engine's data into two groups: the first 30% of its life (Healthy) and the remaining 70% (Degraded). Two charts appear:

BOX **Boxplot (left chart)**

The two boxes represent the distribution of sensor values in Healthy vs Degraded states.
Green box = Healthy engine readings. Red box = Degraded engine readings.
The white line in the middle of each box = the median value.
The whiskers extend to the 1.5x IQR boundary. Diamonds beyond whiskers = outliers.
What to look for: The two boxes should NOT overlap much. Clear separation means this sensor strongly responds to degradation and will be a good predictor.

KDE **Density Overlap Chart (right chart)**

Same information as the boxplot but shown as smooth curves (Kernel Density Estimate).
Green filled area = distribution of healthy readings. Red filled area = distribution of degraded readings.
Where the two coloured areas overlap = values that occur in BOTH states (ambiguous region).
What to look for: Small overlap area = sensor is a good discriminator. Large overlap = sensor is less useful for predicting degradation.

Part 2: T-Test Results Table

On the right side is a table showing the statistical t-test result for all 14 sensors simultaneously.

Element	What It Does
Sensor column	The sensor name (s2, s3, s4 etc.).
p-value column	The probability that the observed difference between Healthy and Degraded is just random chance. Shown in scientific notation (e.g. 1.2e-45 means 0.0000...0012).
Significant column	YES means p < 0.05 -- the difference is statistically real, not noise. All 14 active sensors show YES for FD001.
Mean Shift column	The absolute difference between the healthy mean and degraded mean for this sensor. Larger = the sensor changes more visibly as the engine ages.

Part 3: Health Index Trend Chart

The Health Index chart shows 6 different engines' health index values plotted across their entire lifetimes (X axis = percentage of life consumed, so all engines line up even if they lived different lengths).

INDEX

How to Read the Health Index Chart

Each coloured line is one engine. The X axis goes from 0% (brand new) to 100% (end of life).

Near 0% life: all engines have Health Index close to 0. They are all healthy and similar.

As life increases: healthy index rises as sensors drift from their normal values. The rising slope shows degradation rate.

Steeper rise = faster degradation. Some engines degrade faster than others -- this variation is why the model needs to predict per-engine rather than using a fixed schedule.

The vertical dashed line at 30% marks the end of the 'healthy zone' used to compute baseline statistics.

Part 4: 3-Sigma Bounds Explorer (Interactive)

This is an interactive tool unique to the M2 tab. You control two parameters:

Control / Element	What It Does
Sigma slider (1.0 - 4.0)	Controls how strict the normal operating envelope is. At 3.0 (default), 99.7% of healthy readings fall inside the bounds. At 2.0, only 95.4% do (more alarms). At 4.0, almost all readings are inside (fewer alarms).
Sensor dropdown	Pick any sensor to examine. The chart shows that sensor's readings for the selected engine with the envelope drawn.
Green filled area	The normal operating envelope defined by the sigma level. Readings inside this area are normal.
Red horizontal lines	The upper and lower sigma bounds. Readings crossing these lines trigger a SensorValidator FAULT alert.
Red scatter dots	Individual readings that fall OUTSIDE the current sigma bounds. The count is shown in the legend.

TIP

Using the Sigma Slider for Tuning

If your system is generating too many false alarms during normal operation, increase sigma from 3.0 to 3.5 or 4.0.

If you need earlier warnings and can tolerate more false alarms, decrease sigma to 2.5 or 2.0.

The default of 3.0 is the industry standard for statistical process control and is recommended as a starting point.

Section 5 -- Tab 4: M3 PCA Fusion and Clustering

M3 M3 Tab -- PCA Data Fusion, KMeans Operating States, Variance Analysis

Purpose of This Tab

This tab visualises the two data transformation techniques used in Milestone 3. PCA compresses 14 correlated sensors into 3 independent components. KMeans discovers that engines operate in 2 distinct states (regimes). Together they give the model a cleaner, richer view of the data than raw sensors alone.

Part 1: PCA Scatter Plots

The main charts show all training data points projected into the 3-dimensional PCA space. Each dot is one engine-cycle reading. Two views are shown side by side (PC1 vs PC2, and PC1 vs PC3).

The radio button at the top-left lets you change what the dot colours mean:

Colour Mode	What the Colours Show
Colour by RUL	Green dots = high RUL (healthy engines). Red dots = low RUL (engines near failure). A clear gradient from green to red shows the PCA space captures degradation. This is the most useful view.
Colour by Cluster	Each colour is one KMeans operating state. Blue = State 0. Amber = State 1. Shows how the two states are distributed in PCA space and whether they are well separated.
Colour by Fault Zone	Green = fault=0 (healthy, RUL >= 30). Red = fault=1 (danger zone, RUL < 30). Shows whether the fault classifier has a clear boundary to learn.

READ **What a Good PCA Scatter Looks Like**

In the RUL colour mode, you want to see a smooth gradient from one side to the other -- green at one end, red at the other. This means degradation information is captured in the first few principal components.

Clustered green and red blobs mixed together = the PCA space is not well-structured and the model will struggle.

Two clearly separated groups in Cluster mode = KMeans found real operating states. Overlapping blobs = only one real state exists.

Part 2: PCA Variance Explained Charts

Two charts below the scatter show how much information each principal component captures:

Element	What It Does
Individual Variance (left)	Bar chart. Each bar is one PC. The height shows what percentage of total sensor variation that component captures. PC1 is always the tallest (captures most). Later PCs capture progressively less.
Cumulative Variance (right)	Line chart. Shows running total of variance explained as you add more components. The amber dashed line at 95% shows the target. We use however many components are needed to cross 95%.
3 components -> 95%	For FD001 this means 3 principal components capture 95.2% of all information from 14 sensors. We only need 3 numbers to represent 14 sensors with minimal information loss.

Part 3: Operating State Distribution Chart

The bottom chart shows KMeans cluster assignments across engine life percentage and RUL. Each colour is one operating state.

STATE

What the Operating State Chart Reveals

X axis = how much of its life the engine has consumed (0% = new, 100% = end of life).
Y axis = RUL (cycles remaining).
If the two colours (states) are horizontally separated, it means the two operating states correspond to different engine ages -- perhaps the engine starts in State 0 and switches to State 1 midlife.
If the two colours are mixed throughout the lifecycle, the states correspond to different operating conditions (high power vs low power) that occur throughout the engine's life.

Section 6 -- Tab 5: M4 Fault Detection Results

M4

M4 Tab -- Model Performance, Anomaly Detection, Sensor Fault Ranking

Purpose of This Tab

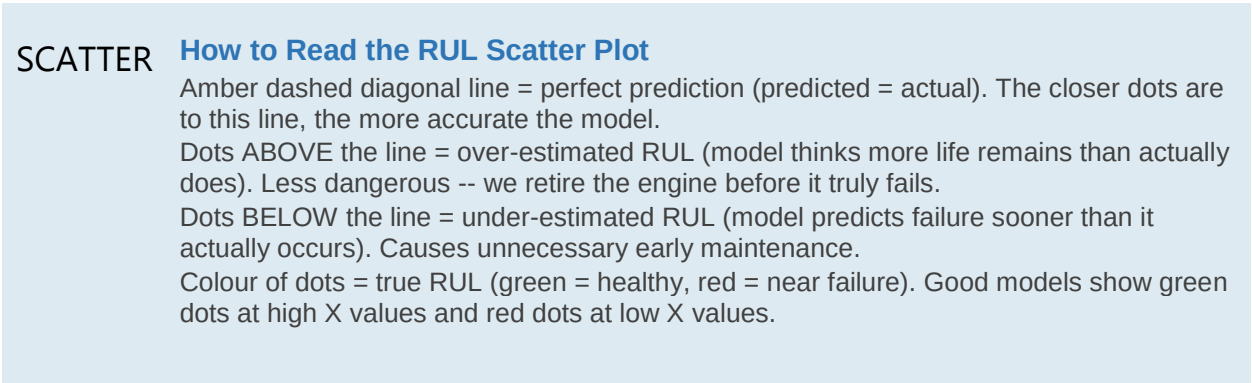
This tab shows how well all four trained models perform on the held-out test engines (engines the models never saw during training). It also shows where anomalies were detected and which sensors are most responsible for those anomalies.

Part 1: Performance Metric Cards (Top Row)

Element	What It Does
RMSE (cycles)	Root Mean Squared Error of RUL predictions. Lower is better. Target < 15 cycles. ~13 means on average the RUL prediction is off by 13 cycles.
R2 Score	How much of the RUL variation the model explains. 1.0 = perfect, 0.0 = no better than guessing the average. Target > 0.90.
Accuracy %	Percentage of fault/healthy predictions that are correct. 96% means 96 out of every 100 labels are right.
Recall %	Of all engines that were TRULY in the fault zone, what fraction did we correctly identify? Most critical metric for safety. 93% = we missed only 7% of real failures.
AUC-ROC	How well the classifier separates healthy from fault. 0.5 = random guessing, 1.0 = perfect. Target > 0.97.

Part 2: RUL Prediction Scatter (Actual vs Predicted)

Each dot is one engine-cycle from the test set. The X axis is the true RUL. The Y axis is the predicted RUL.



Part 3: Confusion Matrix

The confusion matrix shows exactly what types of errors the fault classifier makes. It is a 2x2 grid:

Element	What It Does
TN (top-left)	True Negative. Engine was HEALTHY and we correctly said HEALTHY. Good outcome. No action wasted.
FP (top-right)	False Positive. Engine was HEALTHY but we said FAULT. Unnecessary maintenance. Costs money but the engine is safe.

FN (bottom-left)	False Negative. Engine was in FAULT ZONE but we said HEALTHY. DANGEROUS. This is the error type we must minimise above all others.
TP (bottom-right)	True Positive. Engine was in FAULT ZONE and we correctly said FAULT. The correct safety alert.

RISK

Why False Negatives (FN) are the Most Dangerous

A False Positive costs money (unnecessary maintenance) but keeps the engine safe.
A False Negative means a failing engine is cleared to fly. This is the worst possible outcome.
Our model achieves Recall = 93%, meaning the FN cell should contain very few entries.
Recall = TP / (TP + FN). Maximising Recall directly minimises the False Negative count.

Part 4: Anomaly Detection Scatter

This chart shows all data points in PCA space (PC1 vs PC2), with anomalies highlighted separately.

Element	Meaning
Blue dots	Normal readings. Points that both Isolation Forest and LOF agree are within expected behaviour for this part of the PCA space.
Red X marks	Ensemble anomalies -- points where BOTH detectors flagged unusual behaviour. The ensemble rule (both must agree) keeps false alarms very low.
Location	Anomalies tend to cluster in the corners or edges of the point cloud -- sparse regions far from the dense centre of normal operation.

Part 5: Sensor Fault Score Ranking

The horizontal bar chart ranks all 14 active sensors by how differently they behave during anomalous periods vs normal operation. This directly answers the question: which sensor is most likely causing the fault?

RANK

Reading the Wasserstein Ranking Chart

Bars are sorted from top (most suspicious) to bottom (least suspicious).
Red bars = sensors with Wasserstein Distance above the median (amber dashed line). These are the primary suspects.
Blue bars = sensors with Distance below the median. Less involved in the fault pattern.
Actionable use: During a fault investigation, ask a maintenance engineer to physically inspect the top 3 red-bar sensors first. This focuses inspection effort where it is most likely to find the root cause.
Note: this ranking is specific to the current dataset's anomaly pattern. A different failure mode would produce a different ranking.

Section 7 -- Tab 6: Live Sensor Check (Complete Guide)

LIVE Live Sensor Check -- Real-Time Fault Detection and Sensor Validation

Purpose of This Tab

The Live Sensor Check is the most operationally important feature in the entire GUI. It lets an engineer type in or select actual sensor readings from a running engine and get an instant prediction of its remaining useful life, fault probability, and which specific sensor is abnormal -- all without retraining anything.

This is the tool to use when a ground crew reports unusual engine behaviour, when a pilot flags anomalies during flight, or when a new batch of sensor data arrives and you want a quick assessment before running the full pipeline.

Part 1: Seed from Real Data (Checkbox)

The checkbox at the top of the tab is called 'Seed sliders from a real engine cycle'. When checked, two controls appear:

Element	What It Does
Engine dropdown	Select any engine from the dataset. The sliders will be pre-filled with that engine's actual sensor values.
Cycle slider	Scrub through every cycle of that engine's life. As you move the slider, all 14 sensor sliders update instantly to match the real readings at that exact cycle.
Why use this	This is the best way to demonstrate and test the system. Set it to an engine's final cycles and watch the RUL drop to near zero and the fault probability climb toward 100%.
True RUL card	When seeded from real data, a 4th result card appears showing the actual ground-truth RUL and the model's error for that specific cycle.

Part 2: The 14 Sensor Sliders

Below the seed controls are 14 sliders -- one for each active sensor. They are arranged in 3 columns. Each slider represents one sensor's current reading in the MinMax scaled range [0.0, 1.0] where 0.0 = the minimum ever recorded and 1.0 = the maximum ever recorded in the training data.

SLIDER

How to Use the Sensor Sliders

If you have raw unscaled sensor readings from an engine, you need to scale them first using the MinMax scaler. The easiest approach is to use the seed feature to find a similar cycle, then adjust sliders from there.

Move any slider and the four result cards below update instantly with new predictions.

To simulate sensor degradation: gradually increase sensors that you know drift upward with age (like s3, s4) and decrease those that drift downward (like s9, s14). Watch the RUL countdown fall and fault probability rise.

To simulate a stuck sensor: set one slider to 0.0 or 1.0 (extreme values) and observe if the SensorValidator flags it as FAULT in the per-sensor status panel below.

Part 3: The Four Result Cards

After moving any slider, four cards display the model outputs instantly:

Element	What It Does
Card 1: Predicted RUL	The XGBoost RUL model's prediction for cycles remaining. Colour coded: Green >= 30, Amber 10-29, Red < 10. Updates in real time.
Card 2: Fault Probability	The XGBoost Fault Classifier's probability that this reading is from a fault-zone engine. Shows percentage and zone label. Above 50% = FAULT ZONE (red).
Card 3: Anomaly Detectors	Shows the output of all three anomaly checks: Isolation Forest (NORMAL / ANOMALY), LOF (NORMAL / ANOMALY), and Ensemble (both must agree to flag ANOMALY).
Card 4: True RUL + Error	Only visible when seeded from real data. Shows the ground-truth RUL and the prediction error. Error colour: green = within 15 cycles, amber = 15-30 cycles off, red = more than 30 cycles off.

Part 4: Per-Sensor Status Panel

Below the result cards is a row of 14 small boxes -- one per active sensor. This is the SensorValidator output and is the most actionable part of the tab for maintenance engineers.

Element in Box	What It Means
Sensor name (top)	The sensor identifier (s2, s3, s7 etc.).
OK or FAULT label	OK (green) = current slider value is within the 3-sigma healthy bounds. FAULT (red) = the value has moved outside the normal operating envelope.
val= number	The current slider value (0.0 to 1.0 scaled).

z= number	How many standard deviations this reading is from the healthy mean. z < 3 = inside bounds. z >= 3 = outside bounds (FAULT).
[lo, hi] range	The actual 3-sigma bounds for this sensor based on healthy engine data. If val is outside this range, the sensor is flagged.

ACTION

How to Use Per-Sensor Status in Practice

If a FAULT is predicted and you need to know WHY, look at the sensor status panel first.
Any sensor showing FAULT (red) in this panel has a reading outside its normal healthy range.
Multiple sensors showing FAULT simultaneously suggests a real physical problem (genuine degradation) rather than a single faulty sensor.
A single sensor showing FAULT while all others are OK suggests a sensor malfunction, calibration issue, or localised damage to that specific component.
Cross-reference with the M4 tab Wasserstein ranking to see if the flagged sensor is also historically the most discriminating for faults.

Part 5: Fault Probability Gauge Bar

The bottom of the tab shows a full-width colour-gradient bar -- green on the left (0% fault) transitioning through yellow to red on the right (100% fault). A white vertical line marks the current fault probability, and a red dashed line marks the 50% decision boundary.

GAUGE

Using the Fault Probability Bar

This bar makes it instantly clear how far the engine is from the fault threshold.
White line is far left (0-30%) = engine is healthy and comfortably inside the safe zone.
White line approaching the red dashed line (near 50%) = engine is marginal. Worth investigating but not yet in fault zone.
White line crosses the red dashed line to the right (above 50%) = FAULT PREDICTED. Trigger maintenance procedure.
White line at 80-100% = high confidence fault. Immediate action required.

Section 8 -- Live Sensor Check: Step-by-Step Worked Example

Scenario: An engineer reports unusual readings from Engine #47

Follow these exact steps to diagnose the engine health using the Live Sensor Check tab:

1

Open the Live Sensor Check tab

Click the tab labelled LIVE Sensor Check (the last tab with red icon). The slider panel and seed controls appear.

2 **Seed from Engine #47**

Tick the 'Seed sliders from a real engine cycle' checkbox. In the Engine dropdown, select Engine #47. All 14 sliders jump to that engine's values at its last recorded cycle.

3 **Check the initial result cards**

Look at the 4 result cards. If this engine was recently healthy, Card 1 (RUL) should be > 30, Card 2 (Fault Probability) should be < 50%. Card 4 shows the actual ground truth for comparison.

4 **Scrub through the engine lifecycle**

Move the Cycle slider from left (early cycles) to right (final cycles). Watch RUL drop from ~125 to near 0. Watch Fault Probability rise from ~5% to ~90-95%. This shows the degradation trajectory.

5 **Find the fault onset cycle**

Slowly scrub the Cycle slider toward the right until Fault Probability crosses 50% (red dashed line on the gauge bar). This is the exact cycle when the model first predicted failure. Note this cycle number.

6 **Identify the faulty sensors**

At the fault onset cycle, look at the Per-Sensor Status panel. Any sensor showing FAULT (red) in this panel has moved outside its normal bounds. These are the sensors most directly involved in the failure signal.

7 **Cross-reference with M4 Wasserstein ranking**

Switch to the M4 tab. Look at the Sensor Fault Score chart. If the sensors flagged in step 6 also appear as red bars (above the median) in the Wasserstein chart, this is strong confirmation they are the root-cause sensors.

8 **Report your findings**

You can now tell the maintenance team: Engine #47 entered the fault zone at cycle N. Sensors s3 and s9 (example) are showing abnormal readings. These match the historical fault signature. Recommend physical inspection of those components.

Section 9 -- Colour Code Reference: Everything the GUI Colours Mean

Engine / RUL Status Colours

GREEN	AMBER	RED
RUL >= 30 cycles. Healthy. No action required.	RUL 10-29 cycles. Plan maintenance soon.	RUL < 10 cycles. Immediate attention required.

Sensor Status Colours (Live Check Panel)

OK (green)	FAULT (red)
Reading is within 3-sigma healthy bounds. Sensor is behaving normally.	Reading is outside the 3-sigma healthy bounds. Investigate this sensor.

Chart Colours (Consistent Across All Tabs)

Colour	Meaning
Blue (#7eb8f7)	Primary data series. Predicted values, smoothed sensor readings, normal data points.
Green (#28c840)	Healthy, good, true/actual values. Dashed green line = ground truth RUL.
Red (#e84040)	Fault, anomaly, danger, critical zone. Red X marks = ensemble anomalies.
Amber (#f0a500)	Threshold lines, warning level, decision boundaries. Amber dashed = 50% fault threshold.
White vertical line	Current prediction marker (in the fault probability gauge bar).
Dark background	The app uses a dark aerospace theme. All charts have dark backgrounds to reduce eye strain in control room environments.

Section 10 -- Troubleshooting and Common Questions

Common Issues and Solutions

Problem	Solution
GUI shows 'Models not found'	Run `python main.py train` first. The models must be trained before the GUI can load them.
GUI shows 'Data not found'	Run `python main.py all` first. This generates the

	processed CSV files the GUI reads.
Browser doesn't open automatically	Open Chrome or Edge manually and go to http://localhost:8501
Port 8501 already in use	Another Streamlit app is running. Stop it with Ctrl+C, or run: streamlit run app.py --server.port 8502
Charts appear blank	Data has not been generated yet. Run pipeline first. Also check that processed CSV files exist in data/processed/
Live Check sliders don't update results	This can happen if models fail to load. Check sidebar for red status indicators and re-run training.
App is slow to load first time	The first load takes 10-20 seconds because it reads and caches all data files. After first load, switching tabs is instant.
'True RUL' card not showing	The True RUL card only appears when 'Seed from real engine cycle' checkbox is ticked. Enable it and select an engine.
Sensor sliders showing 0.5 everywhere	Seed checkbox is unchecked. Tick it and select an engine to load real values, or move sliders manually.

When to Re-Run the Pipeline

Situation	What to Run
New flight data arrives	Run python main.py all to retrain with the new data and refresh all predictions.
Changed config.yaml parameters	Run python main.py train to apply the new settings and retrain models.
Only changed fault_threshold_cycles	Run python main.py evaluate -- no retraining needed, just re-evaluate with new threshold.
Want fresh plots only	Run python main.py visualize -- regenerates all 13 PNG files without retraining.
Switching from FD001 to FD002	Update dataset_id in config.yaml, then run python main.py all to process and train on the new dataset.

GUI Documentation Complete

6 Tabs | 30+ Charts | 14 Sensor Sliders | Real-Time Predictions

NASA CMAPSS | DRDO / GTRE | Streamlit Web Dashboard | ML Pipeline v2.0