Turbofan Sensor-FDI-Bench: A Synthetic Dataset for Sensor Fault Detection & Isolation under Degradation and Operating Variability

Aytunc Yildirim[1], Martin Bolemant[2], and Marvin Nöthen[3]

[1,2,3] *DLR—German Aerospace Center, Institute of Propulsion Technology, Cologne, Germany*
aytunc.yildirim@dlr.de
martin.bolemant@dlr.de
marvin.noethen@dlr.de

ABSTRACT

Aircraft engine monitoring relies on sensor measurements to assess gas path condition and to distinguish gradual degradation from abrupt performance changes. In practice, however, sensor signals are influenced simultaneously by the engine state, changing operating conditions, and sensor side effects such as step, drift, random outliers, and measurement noise. This makes it difficult to determine whether an observed deviation originates from the engine, the environment, or the sensing system. For the development and fair comparison of sensor fault detection and isolation methods, a benchmark is required that represents these effects in a controlled and labelled manner. Most publicly available turbofan datasets, however, are primarily intended for remaining useful life prediction and do not provide standardised sensor fault cases for reproducible FDI evaluation. To address this gap, the Turbofan Sensor-FDI-Bench is introduced as a synthetic steady state benchmark dataset generated with a physics based turbofan performance model. The benchmark consists of cruise operating point snapshots and provides, for each flight, environmental conditions, an extended sensor package, and gradual multi component performance degradation. Structured sensor faults with controlled onset and severity are superimposed, including step and drift faults as well as stochastic measurement disturbances. The benchmark is organised as a progressive suite of subsets with increasing complexity, covering fixed and variable operating conditions as well as single fault and multi fault diagnosis settings. For each engine unit, clean reference sensor values are released alongside noisy or faulty measurements, enabling supervised denoising and controlled evaluation of sensor fault diagnosis methods. The resulting benchmark provides a reproducible basis for comparing sensor fault detection and isolation methods under degradation and operating variability.

1. INTRODUCTION

Reliable interpretation of aircraft engine sensor data is central to gas path health monitoring. In turbofan engines, however, measured signals are influenced not only by the engine state itself, but also by the operating point and by imperfections of the sensing system (Fentaye, Baheta, Gilani, & Kyprianidis, 2019). This overlap makes it intrinsically difficult to interpret deviations in pressures, temperatures, shaft speeds, or fuel flow. Gradual degradation is commonly represented through changes in component efficiency and flow capacity, while measurement uncertainty, outliers, and biased sensor behaviour distort the same signals used for diagnosis (Fentaye et al., 2019). Sensor quality is therefore not a secondary issue in engine monitoring, but part of the diagnostic problem itself. Recent studies have investigated denoising and filtering methods for aircraft engine signals, including recursive filtering and neural network based reconstruction approaches (Raikar & Ganguli, 2017; Fentaye et al., 2020; Zhao, Li, & Sampath, 2022). Their development, however, remains constrained by the limited availability of benchmark data that represent physical engine state changes and sensor side disturbances in a controlled, separately traceable manner. In real flight data, such causal separation is rarely available. Public benchmark resources have addressed adjacent problems. The original C-MAPSS datasets were introduced as run-to-failure benchmarks for prognostics (Saxena, Simon, & Eklund, 2008), and later work showed how strongly they shaped data driven prognostics and health management (PHM) research beyond their original use case (Ramasso & Saxena, 2014). N-CMAPSS increased the realism of this benchmark line by simulating full flights under real operating conditions and linking degradation to operational history (Arias Chao, Kulkarni, Goebel, & Fink, 2021). ProDiMES, in contrast, was introduced explicitly as a benchmarking framework for gas path diagnostics and provided a standard problem setup with common evaluation logic (Simon, 2010; Simon, Borguet, Léonard, & Zhang, 2014; Koskoletos, Aretakis, Alexiou, Romesis, & Mathioudakis, 2018). These re-

sources form an important foundation, but they do not provide a directly released fixed dataset suite specifically designed for supervised denoising and structured sensor fault studies under degradation and operating variability. In response, a two stage synthetic data generation workflow is developed. First, clean cruise operating point snapshots are generated using a physics based turbofan performance model that accounts for gradual multi component degradation, engine specific variation in degradation trends, production related offsets, and variable operating conditions. Second, measurement side effects are superimposed on these clean signals as random noise, rare peaks, drift faults, and step faults. On this basis, a progressive benchmark suite is defined, moving from denoising under fixed and variable operating conditions to single fault and multi fault sensor diagnosis settings. Engine disjoint train, validation, and test splits are provided, along with clean reference channels and task specific evaluation protocols.

This paper introduces the Turbofan Sensor-FDI-Bench as a directly released synthetic benchmark dataset suite for supervised denoising and sensor fault diagnosis in turbofan engine health monitoring. Its contribution lies not only in generating synthetic data but also in defining a reproducible benchmark structure that represents degradation, operating variability, and sensor side faults separately and combines them in a controlled manner. The resulting benchmark is intended to support clearer side by side evaluation of denoising and sensor FDI methods than is currently possible with existing public resources.

2. SYNTHETIC DATA GENERATION

Synthetic data are needed for several reasons. First, many aircraft engine manufacturers and operators do not share engine sensor data because of proprietary restrictions. Second, even when some data are made available, they are difficult to use for sensor fault research because the labelling is usually not designed for that purpose. A snapshot from a given flight may contain measured sensor values, but these values are influenced simultaneously by operating conditions, the health state of the gas path components, and sensor imperfections. As a result, a deviation observed in one signal cannot be attributed directly to a single cause when only raw flight snapshots are available. This ambiguity is especially relevant for turbofan engines, where the same measured signal can be influenced by multiple mechanisms simultaneously (Fentaye et al., 2019). On the engine side, the snapshot depends on the current operating point and on the health state of the gas path components, which are commonly reflected by changes in component efficiency, $\Delta\eta$, and mass flow capacity, $\Delta\Gamma$, relative to the intact engine (Fentaye et al., 2019). The health state itself may depend on initial production scatter, long term degradation caused by accumulated usage history, and scatter around the mean degradation trajectory due to transient effects. In addition, abrupt changes in component performance may occur, for example, due to foreign object damage, fatigue, or blade fracture. Such abrupt changes are referred to here as component faults (Fentaye et al., 2019). Their effects propagate through the gas path, thereby influencing several sensors simultaneously. This creates a diagnostic problem distinct from sensor fault detection and isolation. In the present work, the focus is on sensor uncertainty and sensor faults rather than on component fault diagnosis. Nevertheless, slow component degradation must still be included, because it is present in real engines from the first flight onward and forms the background over which sensor faults occur. A synthetic data generation workflow enables these effects to be reproduced in a controlled manner. Instead of observing only isolated snapshots with unclear causal structure, one can generate the complete life history of an engine unit under known operating conditions, known component health trajectories, and known sensor side disturbances. This allows studying how different mechanisms appear in the measured signals and separating engine side effects from measurement side effects. The workflow is divided into two clearly separated stages. First, clean engine measurements are generated using a physics-based turbofan performance model under varying operating conditions and evolving component health states. Second, measurement side imperfections are superimposed on these clean signals as random noise, rare outliers, drift faults, and step faults. This separation is important because it preserves the distinction between physical changes in the engine and distortions introduced only by the sensing system.

2.1. Physics-based Engine Simulation

To generate the engine side part of the benchmark, a thermodynamic performance model was developed in the GTlab environment for a high bypass civil turbofan engine class comparable to the thrust levels of the IAE-V2500 (Reitenbach et al., 2020). Layout of the turbofan engine model in GTlab environment is illustrated in Figure 1. The performance model contains modules for the fan, booster, or low pressure compressor (LPC), high pressure compressor (HPC), high pressure turbine (HPT), and low pressure turbine (LPT). Each component is represented by a performance map that can be scaled by health parameters. In this work, the health state of each component is described by an efficiency deviation and a flow capacity deviation relative to the intact reference condition (Fentaye et al., 2019). For engine unit e at flight cycle k, the input vector, $\mathbf{u}$, to the off-design simulation can be written as:

$$\mathbf{u}^{(e)}(k) = \Big[ALT^{(e)}(k),\, Ma^{(e)}(k),\, \Delta T^{(e)}_{\mathrm{ISA}}(k),\, \pi^{(e)}_{\mathrm{EPR}}(k),\, \mathbf{h}^{(e)}(k)\Big]^{\top}. \quad (1)$$

Here, ALT denotes altitude, Ma denotes Mach number, ΔT_{ISA} is the deviation of the outside air temperature from the Inter-

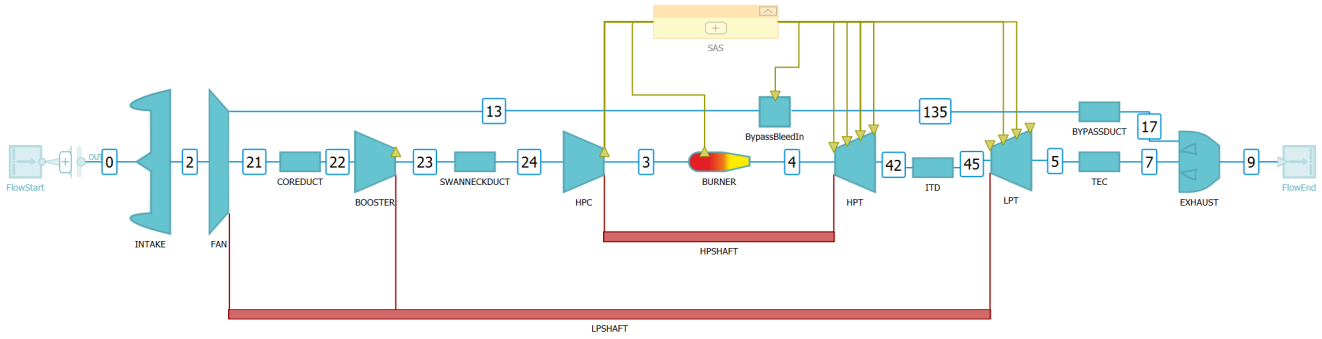

Figure 1. Turbofan layout in GTlab environment.

national Standard Atmosphere, and π_{EPR} is the target engine pressure ratio used to prescribe the power setting. The vector $\mathbf{h}^{(e)}(k)$ contains the component health indices of the engine at flight cycle k. The component health vector is defined as:

$$\mathbf{h}^{(e)}(k) = \Big[\Delta\eta^{(e)}_{\mathrm{FAN}}(k),\, \Delta\Gamma^{(e)}_{\mathrm{FAN}}(k),\, \ldots, \Delta\eta^{(e)}_{\mathrm{LPT}}(k),\, \Delta\Gamma^{(e)}_{\mathrm{LPT}}(k)\Big]^{\top}. \tag{2}$$

Here, $\Delta\eta^{(e)}_{j}(k)$ denotes the efficiency deviation of component j from the intact reference state, and $\Delta\Gamma^{(e)}_{j}(k)$ denotes the corresponding deviation in flow capacity for the same component. The index j refers to one of the five gas path modules FAN, LPC, HPC, HPT, or LPT. Once the boundary conditions and health indices are specified, an off-design performance calculation is carried out for each flight cycle. In this way, each flight cycle yields a physically consistent engine state and a corresponding clean set of sensor values. The clean sensor vector can be expressed as:

$$\mathbf{y}^{(e)}_{\mathrm{ref}}(k) = \mathcal{M}\big(\mathbf{u}^{(e)}(k)\big), \tag{3}$$

where $\mathcal{M}(\cdot)$ denotes the non-linear thermodynamic off-design model and $\mathbf{y}^{(e)}_{\mathrm{ref}}(k)$ is the resulting clean sensor snapshot. In the released benchmark, one steady state cruise snapshot is published for each flight. Cruise operating point snapshots were chosen because they provide a consistent sampling convention over the life of the engine while still preserving the effects of changing environmental conditions and health state. A central point of the generator is that several different engine side sources of variation are explicitly represented rather than merged into a single uncertainty term. For a generic health quantity $\Delta\chi \in \{\Delta\eta, \Delta\Gamma\}$, the cycle wise health evolution can be described conceptually as:

$$\Delta\chi^{(e)}_{j}(k) = \overline{\Delta\chi}_{j}(k) + \delta\chi^{(e)}_{j,\mathrm{var}}(k) + \delta\chi^{(e)}_{j,\mathrm{prod}} + \varepsilon^{(e)}_{j,\mathrm{comp}}(k). \tag{4}$$

Here, $\overline{\Delta\chi}_{j}(k)$ is the mean degradation trend of component j, $\delta\chi^{(e)}_{j,\mathrm{var}}(k)$ is the engine specific degradation variation around that trend, $\delta\chi^{(e)}_{j,\mathrm{prod}}$ is the production scatter offset for engine unit e, and $\varepsilon^{(e)}_{j,\mathrm{comp}}(k)$ is the flight-to-flight fluctuations. The mean degradation trend is not imposed as a purely generic exponential curve. Instead, baseline degradation laws were constructed from literature data on component efficiency and flow capacity change versus engine usage, digitized from prior turbofan degradation studies and JT9D diagnostics reports (Chatterjee & Litt, 2003; Mathioudakis, Kamboukos, & Stamatis, 2002; Sallee, 1978, 1979). The extracted source points are compiled in Appendix Figure 7. These digitized trends were then fitted as functions of flight usage. The fitted functions form the base degradation curves for efficiency and flow capacity. Degradation variation is then introduced by perturbing the coefficients of the fitted curves. As a result, all engines follow similar long term tendencies, but some degrade faster and others more slowly. Production scatter is represented as an engine specific offset or normalisation shift, reflecting the fact that a component may initially perform slightly better or worse than the nominal reference because of manufacturing variability. Flight-to-flight fluctuations is finally added on a cycle basis to account for deviations around the long term degradation trajectory due to transient effects. This means that the health parameters do not evolve as perfectly smooth monotonic curves. The overall tendency still reflects degradation, but local flight-to-flight improvements or worsening can occur, which is more realistic for cruise snapshot data. Operating condition variability is treated separately from health variation. Even at cruise, the published snapshot may be affected by differences in altitude, Mach number, temperature deviation from the standard atmosphere, and targeted power level. These quantities are therefore varied across flights according to prescribed statistical distributions. Their influence on the sensor values is significant and should not be confused with health related effects. In this respect, the synthetic workflow reproduces both operational variability and gradual engine degradation before any measurement imperfection

is added. Although the generator can also inject component faults, these are excluded from the released benchmark subsets used in this study. The reason is that component faults cause distributed changes across several sensors and therefore constitute a different diagnostic problem. Their implementation is retained in the workflow for future dataset extensions, but the present benchmark is restricted to operating variability, gradual degradation on the engine side, and uncertainty and faults on the sensor side.

2.2. Sensor Uncertainty & Sensor Fault Injection

After the engine model generates clean sensor snapshots, measurement side disturbances and faults are superimposed. This second stage shows that real engine measurements are affected not only by the operating point and gas path conditions, but also by the sensing system itself. In the synthetic data generation workflow, four main mechanisms are considered: random noise, rare peaks, drift faults, and step faults. For sensor i, engine unit e, and flight cycle k, the recorded measurement can be written as:

$$y_{i,\mathrm{meas}}^{(e)}(k) = y_{i,\mathrm{ref}}^{(e)}(k) + \varepsilon_i^{(e)}(k) + o_i^{(e)}(k) + d_i^{(e)}(k) + s_i^{(e)}(k). \tag{5}$$

Here, $y_{i,\mathrm{ref}}^{(e)}(k)$ is the clean sensor value obtained from the thermodynamic model, $\varepsilon_i^{(e)}(k)$ is the random measurement noise, $o_i^{(e)}(k)$ is the rare peak or outlier term, $d_i^{(e)}(k)$ is the drift fault contribution, and $s_i^{(e)}(k)$ is the step fault contribution. Not every term is included in every case. The benchmark subsets activate them according to the selected scenario definition. Random measurement noise is modelled as Gaussian noise. For each sensor, the standard deviation is defined as a percentage of the signal magnitude, so that the noise level scales with the corresponding measurement. This can be written as:

$$\begin{aligned} \varepsilon_i^{(e)}(k) &\sim \mathcal{N}\big(0, \sigma_i^2(k)\big), \\ \sigma_i(k) &= \alpha_i \left| y_{i,\mathrm{ref}}^{(e)}(k) \right|. \end{aligned} \tag{6}$$

Here, $\sigma_i(k)$ is the standard deviation of sensor i at flight cycle k, and α_i is the sensor specific noise fraction. The sensor noise settings, average standard deviation levels $\bar{\sigma}_i$, engineering units, and structured sensor fault identifiers used in the benchmark are summarized in Table 1. The percentage noise values α_i were adopted from ProDiMES for $PS0$, $P2$, WFE, $P023$, $P030$, $T2$, $T023$, $T030$, and $T050$, whereas the values for NL, NH, $P044$, $P050$, and $P134$ were taken from Marinai's thesis (Simon, 2010; Marinai, 2004). In some scenarios, the noise fraction can be gradually increased with the cycle index to reflect the possibility that an otherwise healthy measurement channel becomes slightly noisier over time. This mechanism primarily reduces the signal-to-noise ratio, making small deviations more difficult to isolate. A gradually increasing noise level can become difficult to distinguish from a drift type sensor fault, which would blur the separation between the two effects. To limit the scope of the released datasets, the noise percentage is kept constant throughout the cycle history in this study.

Real engine datasets also contain occasional measurements that do not follow the assumed Gaussian distribution. These rare but large excursions are represented as peak noise events. They are injected only in a small percentage of sampled flight cycles. Their magnitude is obtained by multiplying the nominal sensor standard deviation by a randomly selected factor:

$$\begin{aligned} o_i^{(e)}(k) &\sim \mathcal{N}\big(0, \lambda_i^2(k)\sigma_i^2(k)\big), \\ \lambda_i(k) &\sim \mathcal{U}\big(\lambda_{\min}, \lambda_{\max}\big). \end{aligned} \tag{7}$$

Here, $\lambda_i(k)$ is an outlier multiplier sampled from a uniform distribution, and $\mathcal{U}(\cdot)$ denotes the uniform distribution. In the present workflow, this multiplier is sampled within a predefined range, such as 1 to 10, so that the outlier magnitude is selected from a normal distribution with substantially larger deviation than the nominal random noise. Rare peaks are not treated as structured faults, but they are important because they can bias diagnosis algorithms toward false alarms if robustness against outliers is poor. Subsequently, structured sensor faults with drift or step characteristics can be injected into the measurement readings. Drift faults model sensors whose readings move gradually away from the true value as the engine continues to operate. The total drift magnitude at the end of the simulated life is sampled as:

$$A_{d,i} = s_{d,i}\, \lambda_{d,i}\, \overline{\sigma}_i. \tag{8}$$

Here, $A_{d,i}$ is the final drift amplitude for sensor i, $s_{d,i} \in \{-1, +1\}$ is a randomly selected sign, $\lambda_{d,i}$ is a uniformly sampled drift multiplier, and $\overline{\sigma}_i$ is the average standard deviation level of sensor i listed in Table 1. The drift starts either at a prescribed flight cycle or at a randomly selected initiation cycle, depending on the scenario configuration. Once the drift has been initiated at cycle k_0, it grows monotonically toward its final value. For a simulation ending at cycle K, a normalised progress variable is defined as:

$$\xi(k) = \frac{k - k_0}{K - k_0}, \qquad k_0 \le k \le K. \tag{9}$$

Here, $\xi(k)$ is the normalised time coordinate after drift initiation, k_0 is the drift start cycle, and K is the last simulated cycle. A linear drift is then expressed as:

$$\phi_{\mathrm{drift}}(k) = \xi(k), \tag{10}$$

where $\phi_{\mathrm{drift}}(k)$ is the drift progress function. For an expo-

Table 1. Sensor measurement noise settings and structured sensor fault identifiers used in the benchmark.

Sensor	Description	Unit	$\bar{\sigma}_i$	α_i [%]	Drift Fault ID	Step Fault ID	Fault Magnitude
NL	Low pressure spool speed	1/s	0.04	0.05	7	21	$\pm(1\text{–}10)\bar{\sigma}_i$
NH	High pressure spool speed	1/s	0.10	0.05	8	22	$\pm(1\text{–}10)\bar{\sigma}_i$
WFE	Fuel flow rate	kg/s	0.0024	0.60	9	23	$\pm(1\text{–}10)\bar{\sigma}_i$
$PS0$	Ambient static pressure	Pa	80.0	0.15	10	24	$\pm(1\text{–}10)\bar{\sigma}_i$
$P2$	Fan inlet total pressure	Pa	35.0	0.15	11	25	$\pm(1\text{–}10)\bar{\sigma}_i$
$P023$	LPC exit total pressure	Pa	450.0	0.50	12	26	$\pm(1\text{–}10)\bar{\sigma}_i$
$P030$	HPC exit total pressure	Pa	2000.0	0.20	13	27	$\pm(1\text{–}10)\bar{\sigma}_i$
$P044$	HPT exit total pressure	Pa	600.0	0.25	14	28	$\pm(1\text{–}10)\bar{\sigma}_i$
$P050$	LPT exit total pressure	Pa	170.0	0.25	15	29	$\pm(1\text{–}10)\bar{\sigma}_i$
$P134$	Bypass duct total pressure	Pa	150.0	0.25	16	30	$\pm(1\text{–}10)\bar{\sigma}_i$
$T2$	Fan inlet total temperature	K	0.6	0.16	17	31	$\pm(1\text{–}10)\bar{\sigma}_i$
$T023$	LPC exit total temperature	K	0.5	0.16	18	32	$\pm(1\text{–}10)\bar{\sigma}_i$
$T030$	HPC exit total temperature	K	1.1	0.16	19	33	$\pm(1\text{–}10)\bar{\sigma}_i$
$T050$	LPT exit total temperature	K	4.0	0.50	20	34	$\pm(1\text{–}10)\bar{\sigma}_i$

nential drift, the progress function becomes:

$$\phi_{\text{drift}}(k) = \frac{e^{\beta\xi(k)} - 1}{e^{\beta} - 1}, \qquad \beta \sim \mathcal{U}(\beta_{\min}, \beta_{\max}). \tag{11}$$

Here, β is a shape parameter controlling how strongly the drift accelerates over time. The drift contribution added to the measurement is then:

$$d_i^{(e)}(k) = A_{d,i}\,\phi_{\text{drift}}(k), \qquad k \geq k_0. \tag{12}$$

This means that the measured value already containing random noise and possible outliers is shifted gradually away from its clean reference. Drift faults are diagnostically difficult because they can resemble long term degradation or slowly varying calibration changes if the algorithm does not explicitly separate engine side and sensor side trends. Finally, step faults can also be injected in a structured way within the synthetic data generation workflow. Step faults represent sudden or short term biased shifts in the measured signal. In the workflow, they are applied to a selected sensor starting from a fixed or randomly sampled flight cycle. Step fault magnitude is sampled analogously to the drift amplitude:

$$A_{s,i} = s_{s,i}\,\lambda_{s,i}\,\overline{\sigma}_i, \tag{13}$$

where $A_{s,i}$ is the step fault magnitude for sensor i, $s_{s,i} \in \{-1, +1\}$ is the randomly selected sign, $\lambda_{s,i}$ is a uniformly sampled multiplier, and $\overline{\sigma}_i$ is the corresponding average standard deviation level given in Table 1. Two step fault modes are considered. In the abrupt case, the full fault magnitude appears immediately after the initiation cycle. In the rapid case, the full step is reached over a short number of subsequent flights. This can be expressed as:

$$s_i^{(e)}(k) = A_{s,i}\,\phi_{\text{step}}(k), \qquad k \geq k_0, \tag{14}$$

with

$$\phi_{\text{step}}(k) = 1 \tag{15}$$

for an abrupt step, and

$$\phi_{\text{step}}(k) = \min\left(1, \frac{k - k_0 + 1}{L_s + 1}\right) \tag{16}$$

for a rapid step. Here, $\phi_{\text{step}}(k)$ is the step progress function and L_s is the number of flights over which the rapid step develops. Usually, L_s is set to 3 to 8 flights when generating synthetic data if the rapid step fault is present. Abrupt steps are comparatively easy to detect but can be mistaken for sudden operating point changes if the context is ignored. Rapid steps are more subtle because they do not emerge as a single discontinuity, yet they still reach a biased plateau over a short horizon and therefore differ from long term drift. All these structured sensor fault types are illustrated in Figure 2 in a generic way. These mechanisms introduce distinct diagnostic

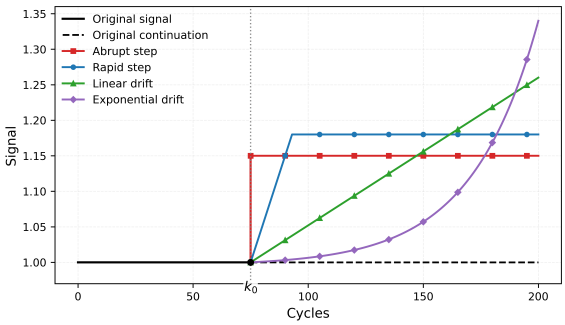

Figure 2. Generic structured fault types on a sensor signal.

difficulties. Random noise obscures weak patterns, rare peaks challenge outlier robustness, drift faults can mimic slow physical trends, and step faults can resemble regime changes or abrupt recalibration effects. By injecting these mechanisms in a controlled and reproducible way, the synthetic workflow makes it possible to study their individual and combined influence on sensor trajectories. At the same time, the benchmark preserves the clean reference values generated by the engine model, so that noisy or faulty measurements can be

compared directly against the corresponding fault free signals. This is particularly useful for supervised denoising and for the controlled evaluation of sensor fault detection and isolation methods.

3. BENCHMARK DESIGN

Public turbofan benchmarks have been valuable for engine health research, but their primary design objectives differ from those of controlled sensor fault benchmarking. The 2008 C-MAPSS challenge data were introduced for data driven prognostics and remaining useful life estimation using noisy steady state snapshots arranged as run-to-failure trajectories (Saxena et al., 2008). As later discussed by Ramasso and Saxena, C-MAPSS became highly influential beyond this original framing, and its subsets should be interpreted as problems of increasing complexity rather than as interchangeable datasets (Ramasso & Saxena, 2014). N-CMAPSS increased this benchmark fidelity by simulating complete transient flights under real flight conditions and linking degradation to operational history (Arias Chao et al., 2021). A different line of work is ProDiMES, which provides a standardised gas path diagnostic benchmark problem and evaluation framework through a steady state fleet simulation environment with configurable operating conditions, degradation, noise, and injected faults (Simon, 2010; Simon et al., 2014). In contrast to directly released fixed dataset suites, ProDiMES is used as a benchmarking framework in which users generate fleets and evaluate blind test cases under a prescribed protocol (Simon, 2010). The benchmark proposed in this work is positioned between prognostics oriented run-to-failure datasets and software based diagnostic benchmarking environments. It is released as a fixed synthetic dataset suite for supervised denoising, sensor fault detection, and isolation under multi component degradation and operating variability. Based on the synthetic data generation workflow described in the previous section, the engine side causality remains explicit while the measurement side disturbances are introduced in a controlled and reproducible manner. The four subsets are organised with increasing task difficulty and are provided with predefined train, validation, and test partitions together with suite level metadata, so that methods can be compared on a common basis without requiring users to generate their own fleets.

The present benchmark adopts one cruise operating point snapshot per flight as its published sampling convention. This choice aligns with earlier steady state benchmark traditions, while keeping the released data compact and directly usable. Finally, the benchmark scope with respect to fault type had to be defined. Although the underlying generator can inject component faults, component fault cases are intentionally excluded from the released benchmark. This scope is a deliberate simplification, yet it still broadly and systematically covers the sensor fault detection and isolation problem in the engine health and management pipeline.

3.1. Structure of Datasets and Challenges

The released benchmark is organised into four suites, DS01-DS04, which should be interpreted as staged benchmark subsets rather than unrelated datasets. Their common scenario structure is summarised in Table 2, and their task roles and sizes are summarised in Table 3. Across all four suites, component degradation, engine-to-engine degradation variation, flight-to-flight fluctuations, production scatter, random measurement noise, and rare outliers remain active. In contrast, operating variability and structured sensor faults are introduced only in the later suites. Component fault injection remains disabled. Each split file is provided as a flat table containing engine and flight identifiers, operating conditions, health indices, measured sensor channels $y_{i,\mathrm{meas}}(k)$, and corresponding reference channels $y_{i,\mathrm{ref}}(k)$. The released sensor package is listed in Table 1 and spans fuel flow, spool speeds, and representative pressure and temperature measurements along the gas path. Although such a complete sensor package is rarely available in operational data, it is suitable for benchmark studies in which denoising or sensor FDI methods are applied upstream of component diagnostics tools. In Table 2, F1 denotes operating variability, F2 component degradation, F3 engine-to-engine degradation trend variation, F4 flight-to-flight fluctuations of the health parameters, F5 production related offsets, F6 random measurement noise, F7 rare outliers, F8 sensor drift faults, and F9 sensor step faults. Tables 2 and 3 show that the benchmark difficulty increases along two controlled directions. The first is the transition from fixed to variable operating conditions in DS02. The second is the transition from no structured sensor faults in DS01–DS02 to single fault settings in DS03 and representative simultaneous fault settings in DS04. This staged organization is preferable to a single mixed dataset because it makes the source of increasing task difficulty identifiable. All suites are distributed with engine disjoint train, validation, and test partitions. For DS03 and DS04, each fault family is represented across the three partitions. For DS03, the single sensor drift and step fault identifiers follow Table 1. In DS04, a multi fault case is identified by combining the constituent single fault IDs into one composite ID. The separator "00" is inserted between the constituent IDs. This keeps the identifier compact while still allowing the underlying drift and step fault components to be recovered directly from the composite ID.

Table 2. Scenario flags of the released benchmark suites.

Suite	F1	F2	F3	F4	F5	F6	F7	F8	F9
DS01	✗	✓	✓	✓	✓	✓	✓	✗	✗
DS02	✓	✓	✓	✓	✓	✓	✓	✗	✗
DS03	✓	✓	✓	✓	✓	✓	✓	✓	✓
DS04	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 3. Overview of the released benchmark suites.

Suite	Primary role	Structured fault organization	Train #	Validation #	Test #
DS01	Denoising benchmark under fixed operating conditions	No structured sensor faults	140	30	30
DS02	Denoising benchmark under variable operating conditions	No structured sensor faults	140	30	30
DS03	Single fault Sensor-FDI	14 drift families and 14 step families 20 engines per family	392	84	84
DS04	Multi fault Sensor-FDI	48 double fault families and 16 triple fault families 12 engines per double fault and 8 engines per triple fault	464	64	176

3.2. Benchmark Use Cases & Evaluation

The released benchmark is intended primarily for supervised denoising and sensor fault detection. For the former, the task is to reconstruct the clean sensor signal from the corresponding noisy measurement by using the provided reference channel as target. For the latter, the task is to decide whether a structured sensor fault is present. In DS03 and DS04, denoising methods may also be used as preprocessing modules before detection or isolation. In that case, however, they should be judged by their effect on downstream diagnostic performance rather than by reconstruction accuracy alone, since an overly aggressive denoiser may suppress diagnostically relevant signal changes (Koskoletos et al., 2018; Zhao et al., 2022).

For the development of a supervised denoising methodology, the most meaningful evaluation unit is the full trajectory of one sensor for one engine. Let $y_{e,i}^{\mathrm{meas}}(k)$ denote the measured value of sensor i for engine e at flight cycle k, let $y_{e,i}^{\mathrm{ref}}(k)$ denote the corresponding reference value, and let $\hat{y}_{e,i}(k)$ denote the denoised output of the method. The trajectory mean absolute error is then defined as:

$$\mathrm{MAE}_{e,i}^{\mathrm{den}} = \frac{1}{K_e} \sum_{k=1}^{K_e} \left| \hat{y}_{e,i}(k) - y_{e,i}^{\mathrm{ref}}(k) \right|, \tag{17}$$

where K_e is the total number of published flight cycles of engine e. The corresponding error of the original noisy observation is:

$$\mathrm{MAE}_{e,i}^{\mathrm{meas}} = \frac{1}{K_e} \sum_{k=1}^{K_e} \left| y_{e,i}^{\mathrm{meas}}(k) - y_{e,i}^{\mathrm{ref}}(k) \right|. \tag{18}$$

Here, $\mathrm{MAE}_{e,i}^{\mathrm{den}}$ measures the residual trajectory wise error after denoising, whereas $\mathrm{MAE}_{e,i}^{\mathrm{meas}}$ measures the original trajectory wise error before denoising. Since the published sensors have different units and scales, these MAE values should first be computed separately for each sensor trajectory. They should not be pooled directly across all rows of the dataset. A more comparable denoising metric is the noise reduction rate. Zhao et al. used this notion to quantify how much of the original error is removed by a denoising method. In the present benchmark, it is defined for one engine and one sensor as (Zhao et al., 2022):

$$\rho_{e,i} = \frac{\mathrm{MAE}_{e,i}^{\mathrm{meas}} - \mathrm{MAE}_{e,i}^{\mathrm{den}}}{\mathrm{MAE}_{e,i}^{\mathrm{meas}}} \times 100\,\%. \tag{19}$$

Thus, $\rho_{e,i}$ should be interpreted as a relative improvement measure rather than as an absolute accuracy score. This percentage style interpretation is also consistent with earlier gas turbine denoising studies, where noise reduction was reported separately for each measurement channel and then summarized across channels (Raikar & Ganguli, 2017; Fentaye et al., 2020). For denoising, the primary evaluation quantity is therefore the engine and sensor specific value $\rho_{e,i}$. Aggregation over the fleet should be treated as a secondary summary step. The sensor average noise reduction rate over the considered split can be written as:

$$\bar{\rho}_i = \frac{1}{N_E} \sum_{e=1}^{N_E} \rho_{e,i}, \tag{20}$$

where N_E is the number of engines in the considered split. The overall mean noise reduction rate of the split is then

$$\bar{\rho} = \frac{1}{N_E N_S} \sum_{e=1}^{N_E} \sum_{i=1}^{N_S} \rho_{e,i}, \tag{21}$$

where N_S is the total number of published sensors. For DS01 and DS02, it is therefore recommended to report $\rho_{e,i}$ as the primary denoising result and to use $\bar{\rho}_i$ and $\bar{\rho}$ only as sensor level and split level summaries. In addition, methods should explicitly state whether the denoising procedure is causal or non-causal, since the use of future snapshots may improve smoothing quality while reducing diagnostic usefulness through time delay (Zhao et al., 2022).

The second primary use case is sensor fault detection. Here, the objective is to determine whether a structured sensor fault has started in an engine trajectory and how early it is detected. Because DS03 and DS04 provide explicit fault activation columns, the task can be formulated in a standardised flight wise manner. DS02 serves as the no-fault reference set, and detection should therefore be evaluated separately on the DS02–DS03 and DS02–DS04 tracks. In both cases, model development should use the training split, threshold

selection should be fixed on the validation split, and final results should be reported only on the test split. For DS03, the column `fault_on` directly indicates whether the structured fault has started. The corresponding binary fault state indicator is therefore defined as:

$$o_e(k) = \texttt{fault\_on}_e(k). \tag{22}$$

For DS04, the benchmark may contain up to 3 structured sensor faults per engine. In that case, the overall binary fault state indicator for detection is defined as:

$$\begin{aligned} o_e(k) = \max\big(&\texttt{fault\_a\_on}_e(k), \\ &\texttt{fault\_b\_on}_e(k), \texttt{fault\_c\_on}_e(k)\big). \end{aligned} \tag{23}$$

Thus, $o_e(k) = 0$ indicates that no structured sensor fault is active at flight cycle k, while $o_e(k) = 1$ indicates that at least one structured sensor fault is active. The true onset cycle of engine e is then:

$$k_{0,e} = \min\{k \,:\, o_e(k) = 1\}. \tag{24}$$

To evaluate a detection method, it should output a flight wise alarm sequence. Let $a_e(k) \in \{0, 1\}$ denote the binary alarm decision of the method for engine e at flight cycle k, where $a_e(k) = 1$ means that the method declares a structured sensor fault to be present. The first standardised metric is the flight wise false positive rate, computed only over healthy flights. For the considered detection track, let $\mathcal{H}$ denote the set of all healthy flights in the corresponding split. In the DS02–DS03 track, this set consists of all DS02 flights together with all DS03 flights for which $o_e(k) = 0$. In the DS02–DS04 track, it consists of all DS02 flights together with all DS04 flights for which $o_e(k) = 0$. The flight wise false positive rate is then defined as:

$$\mathrm{FPR} = \frac{\sum_{(e,k)\in\mathcal{H}} \mathrm{I}(a_e(k) = 1)}{|\mathcal{H}|}. \tag{25}$$

Here, $|\mathcal{H}|$ is the total number of healthy flights and $\mathrm{I}(\cdot)$ is the indicator function, which returns 1 if the condition is satisfied and 0 otherwise. Thus, FPR is the fraction of healthy flights that are incorrectly flagged as faulty. The corresponding false alarm rate is defined as:

$$\mathrm{FAR} = \frac{1}{\mathrm{FPR}}. \tag{26}$$

This definition follows the traditional engine health convention, where FAR is interpreted as the average number of healthy flights required to produce one false alarm (Simon, 2010; Koskoletos et al., 2018). Hence, a larger FAR is better. For example, $\mathrm{FAR} = 1000$ means that one false alarm is produced on average every 1000 healthy flights. To ensure fair comparison between methods, the decision threshold should be selected on the validation split such that:

$$\mathrm{FAR} \geq 1000 \text{ flights}, \tag{27}$$

and then kept fixed for the corresponding test split. If a method does not satisfy this condition on the test split, then it does not satisfy the benchmark's recommended evaluation criterion. The second primary metric is the engine level detection rate. Let $\mathcal{E}_\mathrm{f}$ denote the set of faulty engines in the corresponding test split. For each faulty engine, define the first detection cycle as:

$$k_{d,e} = \min\{k \geq k_{0,e} \,:\, a_e(k) = 1\}, \tag{28}$$

and let $k_{d,e} = \infty$ if the method never raises an alarm after fault onset. The engine level detection rate is then:

$$\mathrm{DR} = \frac{\sum_{e\in\mathcal{E}_\mathrm{f}} \mathrm{I}(k_{d,e} < \infty)}{|\mathcal{E}_\mathrm{f}|}. \tag{29}$$

Here, $|\mathcal{E}_\mathrm{f}|$ is the total number of faulty engines. Therefore, DR gives the fraction of faulty engines that are detected at least once after the true fault onset. The ideal value is $\mathrm{DR} = 1$. The third primary metric is the mean detection latency,

$$L_\mathrm{det} = \frac{1}{N_d} \sum_{e=1}^{N_d} (k_{d,e} - k_{0,e}), \tag{30}$$

where N_d is the number of faulty engines that were successfully detected. Thus, L_det measures the average number of flights required to detect the fault after its true onset. Smaller values are better, and $L_\mathrm{det} = 0$ would correspond to immediate detection at the onset cycle. Accordingly, the fault detection protocol is defined by a flight wise alarm sequence $a_e(k)$, validation based threshold selection under the requirement $\mathrm{FAR} \geq 1000$ healthy flights, and final reporting on the test split. For each track, the primary reported metrics should be FAR, DR, and L_det. Under this protocol, a denoising method used as preprocessing should be assessed by its effect on these downstream detection metrics rather than by reconstruction quality alone.

Beyond these two primary tasks, the benchmark can also support additional studies such as single fault family recognition in DS03 and multi fault attribution in DS04. Nevertheless, the benchmark release defines explicit protocols and evaluation metrics primarily for sensor denoising and sensor fault detection.

4. ENGINE AND FLEET DATA CHARACTERISTICS

This section provides a compact visual characterisation of the released data. The explanatory notebooks released with the datasets can be used to further inspect the fleet data structures and the trajectory format of individual engines. The first aim is to illustrate the inputs to the synthetic data genera-

tion workflow, namely the component health parameters and operating conditions along the snapshot trajectory. Second, measurements are illustrated through residual trajectories for a selected sensor set on one engine and residual deviation distributions for the faulty sensor. Lastly, at the fleet level, the bandwidth of deviations and the median of the measurements are controlled through split wise aggregate trends. In this way, the section complements Sections 2 and 3 by connecting the synthetic generation logic to the actual structure of the published trajectories. Figure 3 shows a representative engine trajectory in terms of component health parameters and operating conditions over the flight cycle. The upper

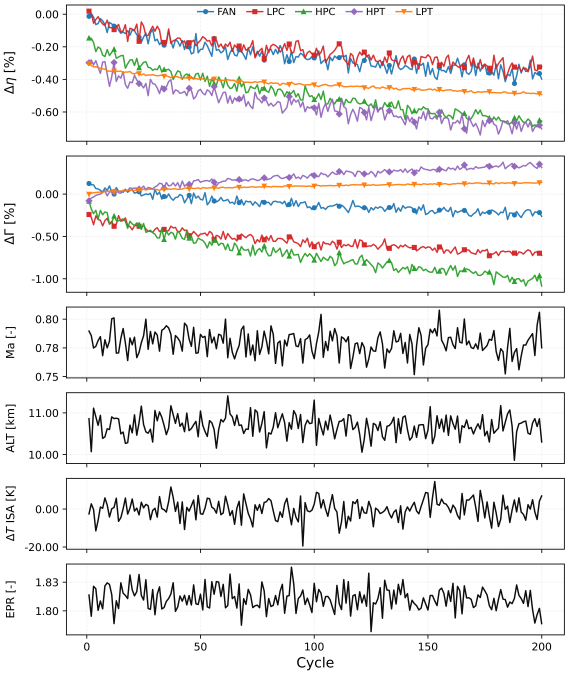

Figure 3. Component parameters and operating conditions.

two panels visualise the component wise efficiency and flow capacity changes, which are stored in the released datasets through the health parameters and are plotted here as percentage deviations, where $\Delta\eta$ denotes the efficiency deviation, and $\Delta\Gamma$ denotes the corresponding flow capacity deviation. The figure illustrates several input properties of the synthetic data generation. First, the long term trend is slow degradation in performance parameters, yet the trajectories are not perfectly smooth due to transient effects during the snapshots. Although each snapshot is at the cruise operating point, there are still flight-to-flight variations in environmental and power level parameters, and the health trajectories also contain local fluctuations due to small transient effects. Also, the starting point for the performance parameters is not necessarily set to 0 due to production scatter effects. Besides, the five modules do not evolve identically. Some components show stronger efficiency decay, whereas others exhibit milder efficiency changes but more pronounced flow capacity shifts. These characteristics vary across engines in the dataset. This characteristic variation in degradation trends is consistent with the benchmark philosophy introduced in Section 3, which states that the background engine state should already be non-trivial before any structured sensor fault is imposed. The lower four plots of Figure 3 show the cycle wise operating point variables Ma, ALT, ΔT_{ISA}, and EPR. Their bounded variations indicate that each cruise snapshot is taken under slightly different environmental and power setting conditions. The baseline values of the operating point variables and their standard deviations are summarized in Table 4. In DS01, where the operating point is fixed, these

Table 4. Baseline values and standard deviations of the cruise operating point variables.

	$\boldsymbol{\mu}$	$\boldsymbol{\sigma}$
ALT	10668 m	250 m
Ma	0.78	0.01
ΔT_{ISA}	0.0 K	5 K
EPR	1.8118	0.01

curves remain constant nominal values, which is not realistic in a real engine snapshot context. In DS02–DS04, by contrast, they fluctuate around the nominal cruise point, thereby introducing an additional source of ambiguity even in the absence of structured sensor faults. Figure 4 shows the residual trajectories of selected measurements for one representative engine. For each sensor i, the plotted quantity is the relative residual between the measured and reference signal,

$$r_i(k) = 100\,\frac{y_{i,\mathrm{meas}}(k) - y_{i,\mathrm{ref}}(k)}{y_{i,\mathrm{ref}}(k)}, \tag{31}$$

where $y_{i,\mathrm{meas}}(k)$ is the measured value and $y_{i,\mathrm{ref}}(k)$ is the corresponding reference value at cycle k. In the example shown, the residuals of WFE, $P030$, and $T050$ remain centred near zero and mainly reflect random measurement noise, with occasional larger deviations due to rare outlier disturbances. On the other hand, the NL signal shows a drift fault starting around cycle 150. Hence, the NL subplot illustrates the nominal sensor side uncertainty background together with a structured sensor fault case. Different sensors exhibit different absolute deviations because the benchmark noise levels are defined relative to the signal magnitude and sensor type. At the same time, the residual trajectories remain bounded in the non-faulty sensors and do not show a persistent systematic departure from zero. This characteristic is the expected behaviour for all sensors of each engine in DS01 and DS02 where no structured sensor fault is present. However, as in this example, at least one sensor shows a structured fault characteristic in DS03 and DS04. The fault onset can also be examined through ridgeline plots of the residuals, as illustrated in Figure 5, which shows the residual distribution of a faulty NL signal across consecutive cycle blocks. Starting

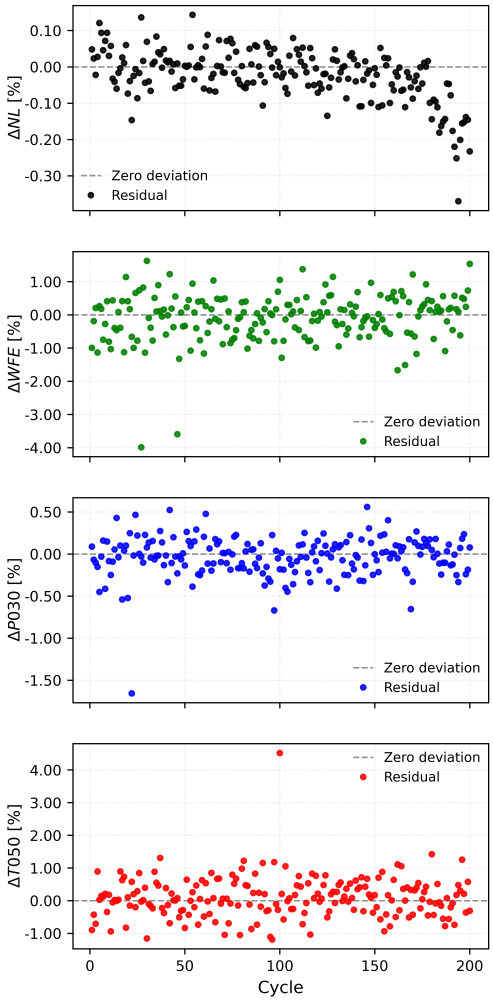

Figure 4. Residual trajectories of selected sensors.

from the same residual definition $r_i(k)$, the trajectory is partitioned into cycle intervals, and for each interval a smooth density estimate is formed,

$$\hat{p}_{i,m}(r) = \frac{1}{n_m h} \sum_{k \in \mathcal{B}_m} K\left(\frac{r - r_i(k)}{h}\right), \tag{32}$$

where $\mathcal{B}_m$ denotes the m-th cycle block, n_m is the number of samples in that block, h is the kernel bandwidth, and $K(\cdot)$ is the kernel function used in the density estimate. The ridgeline representation enables visualisation of how the residual distribution evolves over engine life. In the example shown, the early blocks remain concentrated near zero, whereas later blocks shift progressively towards negative values and broaden. This is the expected signature of a structured sensor fault that becomes increasingly effective with cycle index. Such a representation is useful because it compresses the temporal evolution of a fault into a small number of interpretable distributions. In a no-fault subset such as DS01 or DS02, the ridgelines would remain approximately centred at zero across all cycle blocks, apart from modest widening caused by stochastic measurement effects. In an abrupt step fault case, one would expect a sharper displacement after the fault onset. In a rapid step fault case, the distribution shift would appear over fewer cycle blocks but still more quickly than in a drift case. In DS04, several sensors are faulty, and the corresponding ridgeline plots of those sensors start shifting asynchronously, which is precisely the ambiguity that the multi fault families are designed to create. Lastly, the fleet distribution of the sensor signals is investigated to check whether the splits show balanced behaviour. Hence, Figure 6 complements the single engine views by showing split wise fleet trends for selected sensors in DS03. For each sensor i, split s, and cycle k, consider the cross engine sample of measured values:

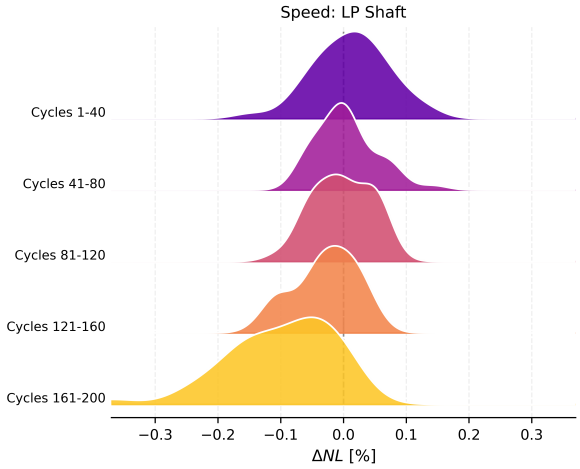

Figure 5. Residual distributions of a faulty sensor across cycle blocks.

$$\mathcal{Y}_{i,s}(k) = \left\{ y^{(e)}_{i,\text{meas}}(k) : e \in \mathcal{E}_s \right\}, \tag{33}$$

where $\mathcal{E}_s$ denotes the set of engines belonging to split s. The plotted centre line is the sample median,

$$\tilde{y}_{i,s}(k) = \text{median}\big(\mathcal{Y}_{i,s}(k)\big), \tag{34}$$

and the shaded band is bounded by the 10^{th} and 90^{th} sample percentiles. Figure 6 provides two complementary views. First, it shows that the train, validation, and test splits are statistically well aligned, since their median trajectories lie close to one another over the full cycle range and their uncertainty bands overlap strongly. This indicates that the benchmark does not introduce an artificial distribution shift between splits. Second, it reveals the natural fleet variability around the central trend. Even when the median behaviour is similar, the percentile bands remain substantial because different engines follow different health trajectories, different operating histories, and, in DS03, different structured fault families. The same type of fleet summary would appear differently across

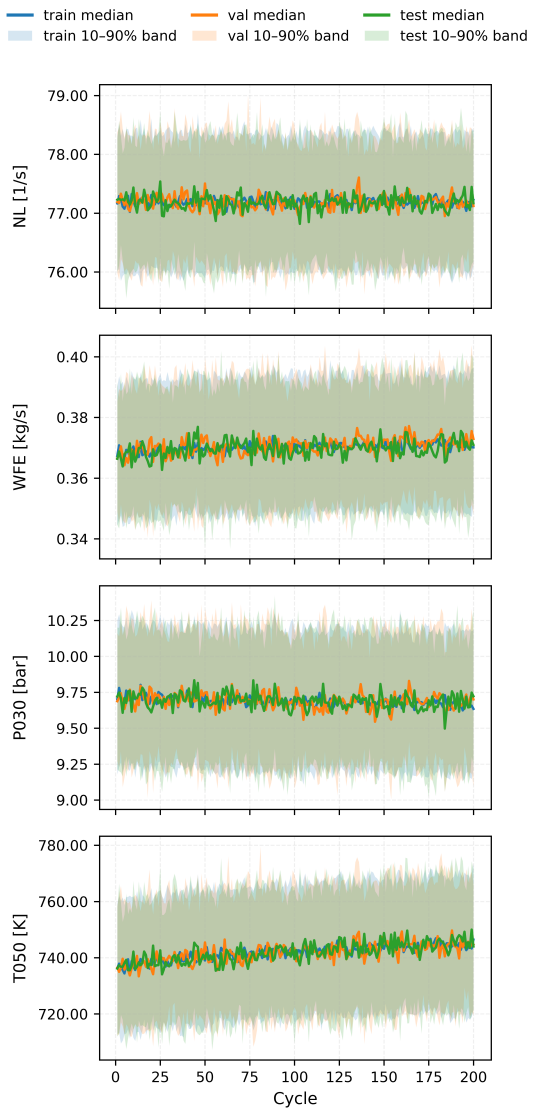

Figure 6. Bandwidth and median trendline of selected sensors in splits of DS03.

the subsets. In DS01, the corresponding bands would generally be tighter because the operating point is fixed and no structured sensor fault is present. In DS02, the bands would broaden because operating variability becomes active. In DS04, the bands can widen further because multi fault families introduce stronger trajectory heterogeneity. Single engine figures show how one engine evolves over the cycle, whereas the fleet level statistics show the envelope within which that evolution should be interpreted. Taken together, Figures 3–6 illustrate the intended hierarchy of the benchmark. At the lowest level, each engine follows its own health and operating trajectory. On top of that trajectory, the measured signals exhibit stochastic residual behaviour and, in the faulted subsets, structured deviations from the clean reference. At the fleet level, these engine specific realizations aggregate into stable split wise statistics with controlled variability.

5. CONCLUSION

This study presented the Turbofan Sensor-FDI-Bench, a synthetic benchmark dataset suite for supervised sensor denoising and sensor fault diagnosis in turbofan engine monitoring. The benchmark was developed to address a specific limitation of the current public literature: widely used turbofan datasets mainly support prognostics, whereas labelled benchmark data for controlled sensor fault studies under degradation and operating variability remain limited. The released benchmark combines four dataset suites with different roles. DS01 and DS02 are intended primarily for denoising studies under fixed and variable operating conditions, respectively. DS03 extends the benchmark to single structured sensor fault cases, and DS04 further introduces simultaneous multi fault cases. Across all four suites, gradual degradation is present, but a single smooth nominal trend does not represent it. Instead, the released trajectories also include engine-to-engine degradation trend variation, production related offsets, and flight-to-flight fluctuations of the health parameters. On the measurement side, random noise and rare outlier disturbances are retained throughout the benchmark. The progression from DS01 to DS04, therefore, does not replace one source of difficulty with another but adds new ambiguity in a controlled way. In the synthetic data generation process, clean cruise snapshots are first generated with a physics based turbofan performance model under degrading component health states and varying operating conditions. Then, measurement side effects are superimposed on these clean signals as random noise, rare peaks, drift faults, and step faults. This separation is central to the benchmark design because it keeps the effect of physical engine state changes distinct from sensor side distortions in the released data. It is therefore possible to study structured sensor faults in a realistic context of degradation and operating variability, rather than in an artificially simplified signal setting. The benchmark was released not only as data files but also as a structured evaluation resource. Train, validation, and test splits are provided for all suites, and task specific protocols were defined for denoising and sensor fault detection. The benchmark is therefore intended to support reproducible method development and more direct comparison across studies. Future work will extend the current release by adding component fault cases, enabling the benchmark to support later studies that examine sensor and engine faults within the same diagnostic framework.

DATA AVAILABILITY

The Turbofan Sensor-FDI-Bench datasets generated and analysed in this study are publicly available on Zenodo at DOI: `10.5281/zenodo.19052907`.

REFERENCES

Arias Chao, M., Kulkarni, C., Goebel, K., & Fink, O. (2021). Aircraft engine run-to-failure dataset under real flight conditions for prognostics and diagnostics. *Data*, *6*(1), 5. doi: 10.3390/data6010005

Chatterjee, S., & Litt, J. (2003). Online model parameter estimation of jet engine degradation for autonomous propulsion control. In *Proceedings of aiaa guidance, navigation, and control conference and exhibit.* Austin, Texas: American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2003-5425

Fentaye, A. D., Baheta, A. T., Gilani, S. I. U., & Kyprianidis, K. G. (2019). A review on gas turbine gas-path diagnostics: State-of-the-art methods, challenges and opportunities. *Aerospace*, *6*(7), 83. doi: 10.3390/aerospace6070083

Fentaye, A. D., Kyprianidis, K. G., Stamoulis, K., Papanikou, M., Apostolidis, A., Plioutsias, A., & Karakoc, H. (2020). An intelligent data filtering and fault detection method for gas turbine engines. *MATEC Web of Conferences*, *314*, 02007. doi: 10.1051/matecconf/202031402007

Koskoletos, A. O., Aretakis, N., Alexiou, A., Romesis, C., & Mathioudakis, K. (2018). Evaluation of aircraft engine gas path diagnostic methods through prodimes. *Journal of Engineering for Gas Turbines and Power*, *140*(12), 121016. doi: 10.1115/1.4040909

Marinai, L. (2004). *Gas-path diagnostics and prognostics for aero-engines using fuzzy logic and time series analysis* (Unpublished doctoral dissertation). Cranfield University.

Mathioudakis, K., Kamboukos, P., & Stamatis, A. (2002). Turbofan performance deterioration tracking using non-linear models and optimization techniques. In *Proceedings of asme turbo expo 2002: Power for land, sea, and air* (pp. 65–73). Amsterdam, The Netherlands: ASME. doi: 10.1115/GT2002-30026

Raikar, C., & Ganguli, R. (2017). Denoising signals used in gas turbine diagnostics with ant colony optimized weighted recursive median filters. *INAE Letters*, *2*(3), 133–143. doi: 10.1007/s41403-017-0023-y

Ramasso, E., & Saxena, A. (2014). Performance benchmarking and analysis of prognostic methods for cmapss datasets. *International Journal of Prognostics and Health Management*, *5*(2). doi: 10.36001/ijphm.2014.v5i2.2236

Reitenbach, S., Vieweg, M., Becker, R., Hollmann, C., Wolters, F., Schmeink, J., … Siggel, M. (2020). Collaborative aircraft engine preliminary design using a virtual engine platform, part a: Architecture and methodology. In *Aiaa scitech 2020 forum.* Reston, Virginia: American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2020-0867

Sallee, G. P. (1978). *Performance deterioration based on existing (historical) data jt9d jet engine diagnostics program* (Tech. Rep. No. NASA-CR-135448; PWA-5512-21). East Hartford, CT, United States: Pratt and Whitney Aircraft Group.

Sallee, G. P. (1979). *Performance deterioration based on in-service engine data: Jt9d jet engine diagnostics program* (Tech. Rep. No. NASA-CR-159525; PWA-5512-35). East Hartford, CT, United States: Pratt and Whitney Aircraft Group.

Saxena, A., Simon, D., & Eklund, N. (2008). Damage propagation modeling for aircraft engine prognostics. In *Proceedings of the international conference on prognostics and health management 2008.* Denver, CO, United States: IEEE.

Simon, D. L. (2010). *Propulsion diagnostic method evaluation strategy (prodimes) user's guide: Technical memorandum* (Tech. Rep. No. NASA/TM-2010-215840). Cleveland, OH, United States: NASA Glenn Research Center.

Simon, D. L., Borguet, S., Léonard, O., & Zhang, X. (2014). Aircraft engine gas path diagnostic methods: Public benchmarking results. *Journal of Engineering for Gas Turbines and Power*, *136*(4). doi: 10.1115/1.4025482

Zhao, J., Li, Y., & Sampath, S. (2022). Convolutional neural network denoising autoencoders for intelligent aircraft engine gas path health signal noise filtering. *Journal of Engineering for Gas Turbines and Power*, 1–22. doi: 10.1115/1.4056128

BIOGRAPHIES

Aytunc Yildirim received the B.Sc. degree in mechanical engineering from Bogazici University, Turkey, in 2022, and the M.Sc. degree in simulation sciences from RWTH Aachen University, Germany, in 2024. He is currently a scientific researcher at the German Aerospace Center (DLR), Institute of Propulsion Technology. His current research interests include aircraft engine health monitoring, gas path diagnostics, measurement uncertainty, sensor fault diagnosis, and propulsion system preliminary design.

Martin Bolemant received the Diplom Ingenieur degree in physical engineering from Technische Universität Berlin, Germany, in 2008, and the Ph.D. degree in physical sciences from Technische Universität Berlin, Germany. He is currently a scientific researcher at the German Aerospace Center (DLR). His current and previous research interests include aerospace propulsion, gas turbines, condition monitoring, and optimization.

Marvin Nöthen received the B.Eng. degree in Computer Science from DHBW Mannheim, Germany, in 2019, and the M.Sc. degree in Computer Science from the University of Koblenz, Germany, in 2024. He is currently a research scientist at the German Aerospace Center (DLR), Cologne, Germany. He is a core developer of GTlab and has contributed to several related software modules for gas turbine preliminary design. His research focuses on gas turbine performance modeling, inverse gas path analysis, and health monitoring.

APPENDIX

Figure 7 compiles the digitized literature points used to define the baseline degradation laws for the five gas path modules. The top row shows efficiency change and the bottom row shows flow-capacity change as functions of flight-cycle usage.The source points in Figure 7 were digitized from (Chatterjee & Litt, 2003; Mathioudakis et al., 2002; Sallee, 1978, 1979).

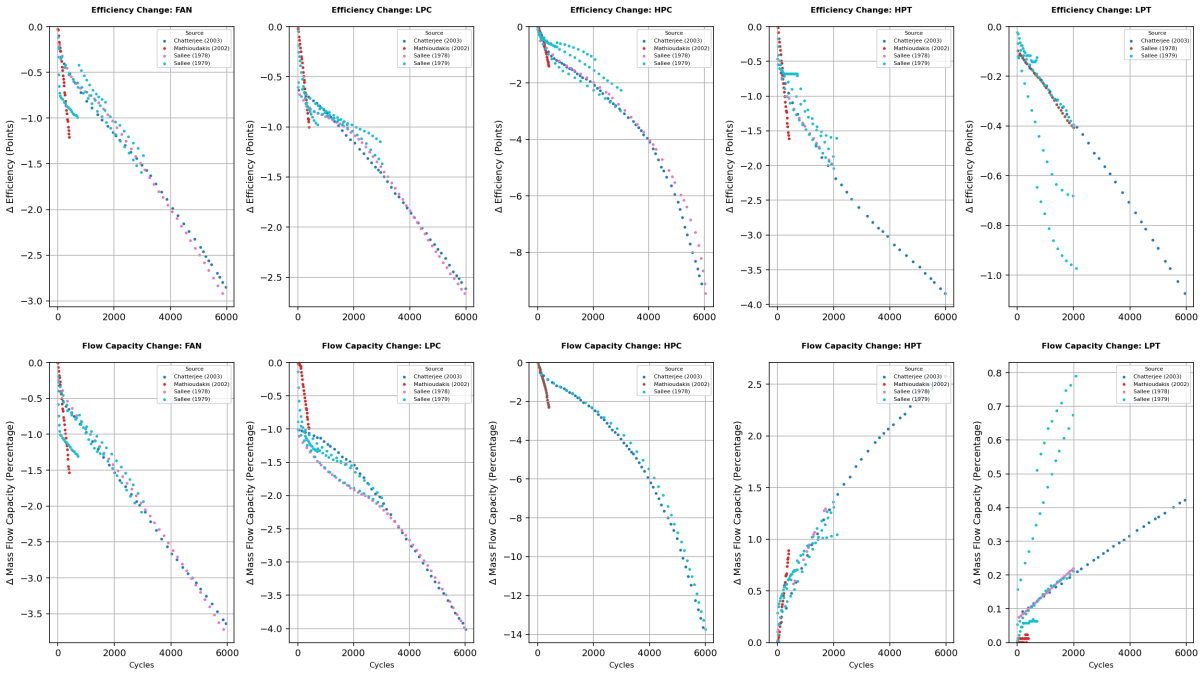

Figure 7. Digitized literature data used to define the baseline degradation trends for component efficiency and flow capacity evolution.