Uncertainty-Aware and Risk-Controlled Identification of Abnormal Parametric Changes in Space Launcher Electrical Valve Actuators

Luis Basora[1], Julien Demange-Chryst[1], Sébastien Priotto[2], Mohammad Ghousein[2], and Serge Le Gonidec[2]

[1] *DTIS, ONERA, Université de Toulouse, 31000, Toulouse, France*
luis.basora@onera.fr
julien.demange-chryst@onera.fr

[2] *ArianeGroup SAS, Forêt de Vernon, 27208 Vernon, France*
sebastien.priotto@ariane.group
mohammad.ghousein@ariane.group
serge.le-gonidec@ariane.group

ABSTRACT

In the context of health monitoring for the next generation of reusable space launchers, this work presents an uncertainty-aware method for detecting and diagnosing off-nominal parameter variations in the electrical system that drives engine valves. The study relies on data generated from a physics-based model, where deviations of nine key parameters simulate realistic faults and degradations.

The proposed pipeline combines domain-driven segmentation to isolate valve-motion intervals, automated statistical feature extraction, and multiclass gradient-boosting-based classification, together with out-of-distribution detection using an isolation forest. To enable uncertainty-aware decisions with user-defined confidence levels, all predictive stages are calibrated through a learn–then–test risk-control framework, providing finite-sample guarantees for out-of-distribution rejection, fault detection, and diagnosis via prediction sets.

Numerical results on the available data demonstrate the effectiveness of the pipeline and an improvement over our previously published approach. However, a class-separability analysis reveals intrinsic limitations of the available signals for near-nominal fault classes, underscoring the need for improved observability or alternative modeling assumptions in future real-data deployments.

1. INTRODUCTION

This research is part of a European effort to develop a health monitoring system (HMS) for the next generation of reusable space launchers. Current European launch systems primarily rely on post-flight telemetry analysis because they are expendable and have limited onboard resources. Future reusable launchers, however, will require advanced onboard monitoring capabilities. Such an HMS is essential to ensure high levels of safety and reliability while reducing operational costs, and must support both in-flight and post-flight decision-making, in line with established practices in aeronautics (Férard, Le Gonidec, Galeotta, Oriol, & Dreyer, 2021).

Within this context, electrical valve actuators are critical components of rocket engines, as they regulate the flow of propellants and other essential fluids under demanding operational and environmental conditions. Degradations or faults affecting these actuators can directly impact engine performance and mission safety. In practice, many actuator faults manifest as deviations of underlying physical parameters—such as electromagnetic, mechanical, or load-related properties—from their nominal values. Identifying such abnormal parametric changes therefore provides a physically grounded and early form of fault detection and diagnosis.

In this work, faults are modeled as abnormal changes in physical parameters of the electrical valve actuator. These parameters include the main magnetic flux, mean inductance, saliency level, and electrical resistance of the electromagnetic subsystem, as well as the actuator and valve inertia, viscous and dry friction, external load torque, and an imbalance parameter capturing structural or electromagnetic asymmetries. Deviations of these parameters from their nominal ranges provide a physically meaningful representation of realistic actuator degradations—such as magnetic aging, winding degradation, increased friction, valve obstruction, or mechanical imbalance. Consequently, fault detection and diagnosis can be naturally formulated as the identification and discrimination of abnormal parametric changes underlying the observed actuator be-

havior.

Fault diagnosis for valve actuators typically relies on multivariate time series derived from electrical and mechanical measurements. Classical approaches include signal-processing and statistical techniques such as spectral analysis, time-frequency representations, and residual-based models, which have been widely applied to electrical machines and electromechanical systems (Benbouzid, 2000; Thomson & Fenger, 2003; Nandi, Toliyat, & Li, 2005). More recently, data-driven and machine-learning approaches have gained prominence, including distance-based methods, ensemble classifiers, and deep-learning architectures that automatically extract discriminative time-series features (Bagnall, Lines, Bostrom, Large, & Keogh, 2017; Ismail Fawaz, Forestier, Weber, Idoumghar, & Muller, 2019; Choi, Yi, Park, & Yoon, 2021). In particular, convolutional and recurrent neural networks, often combined with wavelet or Park-vector representations, have been applied successfully to electrical motor and actuator fault diagnosis (Husebø, Kandukuri, Klausen, Huynh, & Robbersmyr, 2020; Jiménez-Guarneros, Morales-Perez, & de Jesus Rangel-Magdaleno, 2021; Shao, Yan, Lu, Wang, & Gao, 2019; Verma, Henwood, Castella, Malrait, & Pesquet, 2020; Verma, 2023). While these methods offer improved modeling flexibility, their application to safety-critical aerospace systems remains challenging due to constraints on robustness, interpretability, uncertainty quantification, and onboard implementation.

Our previous work (Basora, Bocquet-Nouaille, Robinson, & Gonidec, 2025) addressed the problem considered in this paper using the same simulated dataset together with a deep-learning architecture combined with calibrated gradient-boosting classifiers. In the present study, we improve upon that approach through two major methodological developments. First, we incorporate domain knowledge into the preprocessing stage to isolate valve-motion intervals and extract statistical features tailored to capture parametric variations. Second, we adopt a predictive risk-control framework to select decision thresholds that satisfy target performance criteria, such as a desired precision–recall trade-off.

The primary methodological contribution of this work is the use of the learn–then–test (LTT) framework (Angelopoulos, Bates, Candès, Jordan, & Lei, 2025) for predictive risk control. LTT calibrates predictive algorithms with finite-sample guarantees on user-defined decision risks and performance targets. To the best of our knowledge, such risk-aware guarantees have not previously been investigated in the PHM literature. While conformal prediction has been applied to RUL estimation and fault detection (Javanmardi & Hullermeier, 2023; Robinson, 2026; Yang, Meles, Yilma, & Teshome, 2026; Heddoub, Diallo, Homri, Dantan, & Siadat, 2025; Diallo, Homri, & Dantan, 2025), its guarantees are typically limited to coverage or false positive rates. In contrast, LTT extends these guarantees to arbitrary user-defined risk functions, enabling control over a broader range of operational objectives.

The paper is structured as follows. Section 2 presents the simulated system and data. In Section 3, we describe our proposed modeling approach, highlighting the main steps of the inference process. Section 4 describes the experimental setup, including the dataset and pipeline configuration used to produce the results. Section 5 evaluates the performance of the configured framework using the simulated data. Section 6 discusses class-separability issues, compares the performance of the described method with our previous one, and addresses operational deployment. Section 7 summarizes the key findings, discusses limitations, and outlines potential directions for future research.

2. SIMULATED SYSTEM AND DATA

We consider an electrically actuated valve, driven by a three-phase motor and a closed-loop controller that tracks a prescribed position reference. The mechanical subsystem converts motor torque into a valve opening angle (e.g., from $0°$ to $220°$) under a constant load torque. In this work, faults are simulated as abnormal changes of physical parameters of the electromechanical model.

Raw signals. The simulated dataset consists of multivariate trajectories sampled at 1 kHz. Each trajectory contains 15 time-series signals (Table 1), including electrical (voltages and currents in the dq frame) and mechanical (positions, speeds, and torque) variables. The reference d-axis current *Id_Ref* is constant in all simulations and is therefore excluded from further analysis.

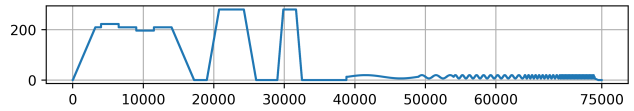


Figure 1. Measured mechanical position along a trajectory.

Derived signals. We define a set of derived signals designed to emphasize fast transients, cross-couplings, and subtle deviations that may be difficult to detect in the raw signals (Table 2). These signals, computed via numerical differentiation and simple normalization of the raw variables, highlight control-loop inconsistencies and electromechanical interactions.

Simulation parameters and fault labels. The simulation is governed by nine physical input parameters. Fault classes are defined by whether one or more of these parameters fall outside their nominal ranges. Table 3 reports the nominal (N) and anomalous (A) ranges along with their associated labels. Labels follow a 9-bit encoding, where each bit indicates

Table 1. Raw simulated signals.

Signal	Symbol	Unit
D-axis reference voltage	Vd_Ref	V
Q-axis reference voltage	Vq_Ref	V
Stator pulsation	Ws	$\mathrm{rad\,s^{-1}}$
D-axis reference current	Id_Ref	A
Q-axis reference current	Iq_Ref	A
D-axis measured current	Id_Meas	A
Q-axis measured current	Iq_Meas	A
Set-point mechanical position	PosM_Set	rad
Reference mechanical position	PosM_Ref	rad
Measured mechanical position	PosM_Meas	rad
Reference mechanical speed	SpeedM_Ref	$\mathrm{rad\,s^{-1}}$
Measured mechanical speed	SpeedM_Meas	$\mathrm{rad\,s^{-1}}$
Reference mechanical torque	TqM_Ref	$\mathrm{N\,m}$
Set-point valve position	PosV_Set	°
Measured valve position	PosV_Meas	°

Table 2. Derived signals capturing cross-couplings and fast transients in the electromechanical valve system.

Derived signal	Rationale
dPos	Position jumps
dSpeed	Acceleration events
dSpeed/TqM	Torque-normalized accel.
dSpeed/Iq	Current-normalized accel.
dSpeed/Speed	Relative accel.
dSpeed/SpeedIq	Speed/current-normalized accel.
dIq	q-current bursts
dIq-Uq	Iq–Vq coupling
dIq-Uq/Iq	Normalized residual
dId	d-current bursts
dId-Ud	Id–Vd coupling
dId-Ud/Iq	Iq-normalized residual
dId-Ud/Iqw	Measured-Iq normalized residual

which parameter is out of range. The nominal class thus corresponds to label 0, while the simultaneous deviation of all nine parameters corresponds to label 511.

The nominal parameter ranges account for production variability. However, actuator impedance, particularly that of the electric motor, is temperature-dependent, an effect not currently represented in the simulations. This assumption is acceptable for the active flight phase considered here, during which the temperature is expected to be stabilized. Pre-flight operating phases, where thermal transients may be more pronounced, will be addressed in future work.

Simultaneous multi-parameter deviations are physically unlikely unless linked to a common-cause failure in a coupled subsystem; only double faults are statistically plausible. The all-deviating case (label 511) was nonetheless included to stress-test the algorithm and verify that it is not misled under this extreme scenario.

Table 3. Simulation input parameters with their nominal (N) and anomalous (A) ranges. Each parameter is assigned a unique bit in a 9-bit fault code, set to 1 when the parameter is out of range (label 0: all nominal; label 511: all deviating).

Parameter	Unit	Range	Label
Flux	Wb	N: 90–110 A: 75–85,115–125	1
Induct.	H	N: 90–110 A: 75–85,115–125	2
Saliency	%	N: 0–3 A: 5–10	4
Resist.	Ω	N: 100–150 A: 50–90,160–200	8
Inertia	$\mathrm{kg\,m^2}$	N: 90–110 A: 50–80,120–150	16
Friction	$\mathrm{N\,m\,s}$	N: 0–110 A: 125–200	32
Dry Frict.	$\mathrm{N\,m}$	N: 0–110 A: 125–200	64
Load Torque	$\mathrm{N\,m}$	N: 0–110 A: 125–200	128
Unbalanced	%	N: 0–2 A: 5–20	256

3. METHOD

In this section, we describe the proposed methodology. As shown in Figure 2, it comprises three main steps: data preprocessing (subsections 3.1 and 3.2), scoring (subsections 3.3 and 3.4), and risk-control-based decisions (subsection 3.5).

3.1. Signal Segmentation

To focus the analysis on the periods most informative for fault detection, each multivariate time-series trajectory is partitioned into segments. This approach leverages domain expertise and restricts learning and detection to intervals when the valve is in motion, ignoring both startup transients and steady-state plateaus.

Formally, let $\mathbf{x}(t) \in \mathbb{R}^d$ for $t = 1, \ldots, T$ denote a multivariate trajectory with d features. We select a feature index $j \in \{1, \ldots, d\}$ and define a binary activity mask

$$m(t) = \begin{cases} 1, & \text{if } |x_j(t)| > \tau, \\ 0, & \text{otherwise,} \end{cases} \qquad t = 1, \ldots, T, \quad (1)$$

where $\tau > 0$ is a fixed amplitude threshold. The selected feature $x_j(t)$ corresponds to the measured mechanical valve speed, SpeedM_Meas(t), and we set $\tau = 1$ to identify periods when the valve is in motion.

A *candidate segment* is defined as a maximal contiguous interval $[s, e)$ such that $m(t) = 1$ for all $t \in [s, e)$. Only segments whose duration satisfies

$$e - s \geq L_{\min} \quad (2)$$

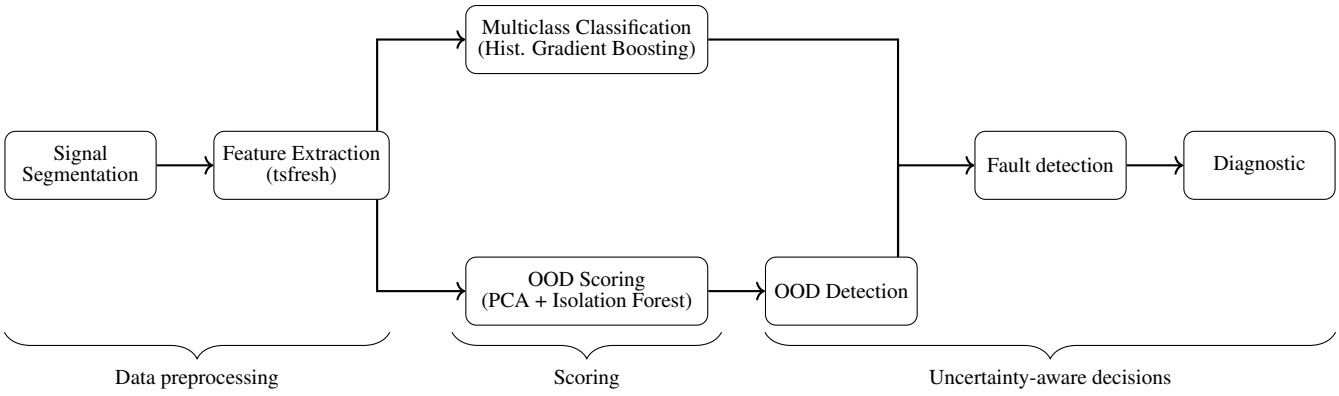


Figure 2. Method pipeline.

are retained, where $L_{\min}$ is a minimum segment length constraint.

To account for brief interruptions caused by noise or threshold crossings, temporally adjacent segments are merged if the gap between them is smaller than a predefined tolerance. Specifically, two consecutive segments $[s_1, e_1)$ and $[s_2, e_2)$ are merged if

$$s_2 - e_1 \leq G_{\max}, \tag{3}$$

where $G_{\max}$ denotes the maximum admissible gap. After merging, the minimum length constraint is enforced again to remove spurious short segments.

3.2. Segment-Level Feature Extraction and Selection

Signal segmentation produces a variable number of segments per trajectory, each with potentially different lengths. To obtain a uniform representation, each segment is converted into a fixed-dimensional feature vector using automated feature extraction. We employ `tsfresh` (Christ, Braun, Neuffer, & Kempa-Liehr, 2018), which provides a diverse set of time-series descriptors capturing temporal dynamics, transient behaviors, and basic spectral characteristics, while supporting relevance-based feature selection and efficient retraining.

Feature extraction. For each segment $\mathbf{x}(t) \in \mathbb{R}^d$, $t = 1, \ldots, T_s$, we compute descriptive statistics with `tsfresh`. The time index is retained to compute order-dependent features, and all features are computed independently for each segment identifier.

By default, we use the *minimal* feature configuration provided by `tsfresh`, which includes basic statistics such as mean, variance, extrema, and simple temporal descriptors. In addition, we augment the default configuration with a set of signal-specific feature calculators available in `tsfresh`. In particular, different feature families are assigned depending on the signal type:

- **Electrical signals**: frequency-domain features (FFT coefficients and aggregated spectra), autocorrelation, peak statistics, quantiles, and change-based measures.
- **Mechanical signals**: trend-based features, peak statistics, quantiles, longest-strike measures, and spectral density estimates.
- **Derived signals** (e.g., errors or numerical derivatives): reduced frequency-domain and autocorrelation features combined with change-based statistics.

This selective assignment limits feature redundancy while retaining physically meaningful descriptors.

Missing feature values resulting from constant or short segments are handled using an imputation strategy provided by `tsfresh`.

Feature filtering and selection. To reduce dimensionality, feature selection is performed in two stages using the training set. First, the built-in relevance filtering procedure of `tsfresh` removes features that are statistically independent of the target labels. Second, a Random Forest classifier is trained to distinguish nominal from faulty segments, and features are ranked according to their importance scores. The top k features are then retained, resulting in a compact and discriminative representation that is applied consistently across all data splits.

3.3. Segment-Level Multiclass Classification

Each segment $\mathbf{x} \in \mathbb{R}^k$ is assigned a fault score using a multiclass classifier. The output of the classifier is a vector of posterior class probabilities

$$\mathbf{p}(\mathbf{x}) = \big(p(y = 0 \mid \mathbf{x}), \ldots, p(y = K \mid \mathbf{x})\big), \tag{4}$$

where K is the number of classes (with 0 denoting the nominal class) and $\mathbf{x}$ is the segment feature vector.

For fault detection, we define a *segment-wise detection prob-*

ability as the posterior probability that a segment is non-nominal,

$$p_{\text{det}}(\mathbf{x}) \triangleq 1 - p(y = 0 \mid \mathbf{x}), \tag{5}$$

which aggregates the posterior mass assigned to all fault classes.

For the underlying model, we use a multiclass `LightGBM` (Ke et al., 2017) classifier, an efficient histogram-based gradient boosting algorithm that integrates naturally with automated tuning tools such as `FLAML` (Wang, Wu, Weimer, & Zhu, 2021). This choice is motivated by our previous benchmark study (Basora et al., 2025), where histogram-based gradient boosting outperformed decision trees, support vector machines, logistic regression, Gaussian naive Bayes, random forests, and k-means.

Hyperparameters are optimized using the zero-shot AutoML strategy of `FLAML` (Kayali & Wang, 2022). The search is initialized from data-dependent configurations obtained through meta-learning, focusing exploration on promising regions of the hyperparameter space. The configuration is then refined adaptively within a fixed computational budget through adaptive hyperparameter optimization based on a validation performance metric.

3.4. Out-of-Distribution Scoring

Each segment $\mathbf{x} \in \mathbb{R}^k$ is standardized and its dimensionality reduced via PCA, retaining enough components to explain 95% of the empirical variance. This yields a compact representation $\widetilde{\mathbf{x}} \in \mathbb{R}^{k'}$, $k' \ll k$, and mitigates the curse of dimensionality that affects distance-based methods in high-dimensional spaces.

An Isolation Forest (Liu, Ting, & Zhou, 2008) is then used to assign an anomaly score to each $\widetilde{\mathbf{x}}$. This method is computationally efficient and does not require labeled OOD samples. It exploits the fact that anomalous observations tend to be isolated with fewer random partitions than in-distribution samples, resulting in higher anomaly scores. The Isolation Forest is fit on the training set, which defines the in-distribution population and includes both nominal and the considered fault classes.

For a given segment with features $\widetilde{\mathbf{x}}$, the Isolation Forest produces a continuous anomaly score via its decision function

$$\text{IF}(\widetilde{\mathbf{x}}) \in \mathbb{R},$$

where positive values correspond to inliers and negative values to outliers in `scikit-learn` (Pedregosa et al., 2011). To simplify interpretation and ensure that higher values indicate stronger deviation from the training distribution, we define the OOD score as

$$s_{\text{OOD}}(\widetilde{\mathbf{x}}) = -\text{IF}(\widetilde{\mathbf{x}}) \tag{6}$$

3.5. Risk-Controlled Decision-Making

Risk control (Bates, Angelopoulos, Lei, Malik, & Jordan, 2021) is a general framework for calibrating predictive models with finite-sample statistical guarantees, so that a user-defined predictive risk stays below a desired level with high probability. Risk control can be viewed as a natural extension of conformal prediction (Shafer & Vovk, 2008), which is primarily designed to ensure coverage guarantees, whereas risk control applies to a broader class of performance metrics beyond coverage. Building on this framework, we translate predictive scores into reliable operational decisions. The outputs produced by the scoring models are processed through three sequential decision stages—OOD rejection, fault detection, and diagnosis—each conditioning on the decisions of the previous ones.

Risk control. In its generic form, risk control considers a calibration dataset $(Z_i, Y_i)_{i=1}^n$, where Z_i denotes the input passed to a decision rule and Y_i is the associated ground-truth label. The input Z_i may be a scalar score, a vector of class probabilities, or any other model-derived score. A parameterized decision rule

$$\mathcal{S}_\lambda(Z)$$

then maps this input to an operational output, such as a binary decision, a class label, or a prediction set. The parameter $\lambda \in \Lambda \subset \mathbb{R} \cup \{+\infty\}$ controls the operating point.

For any decision output $\mathcal{S}_\lambda$, we define a loss $\ell(Y, \mathcal{S}) \in \mathbb{R}_+$ encoding the application-specific error of interest (e.g. misclassification, false alarm, or prediction-set miscoverage $\mathbb{1}\{Y \notin \mathcal{S}_\lambda(Z)\}$). The population risk and its empirical counterpart on calibration data are

$$R(\lambda) = \mathbb{E}[\ell(Y, \mathcal{S}_\lambda(Z))]\,,$$
$$\widehat{R}_n(\lambda) = \frac{1}{n}\sum_{i=1}^{n} \ell(Y_i, \mathcal{S}_\lambda(Z_i))\,.$$

The goal of the risk controller is to select $\hat{\lambda}$ such that

$$\mathbb{P}\big(R(\hat{\lambda}) \le \alpha\big) \ge 1 - \delta,$$

where α is a user-defined acceptable risk level and δ controls the confidence of the finite-sample guarantee. In general, risk-control guarantees are typically formulated under exchangeability of the calibration data.

Learn Then Test. To calibrate λ we use the LTT procedure (Angelopoulos et al., 2025), which requires the data to be independent and identically distributed (i.i.d.), and implies exchangeability. Let λ be a discrete set Λ (if Λ is not discrete,

it needs to be discretized)

$$\Lambda = \{\lambda_1, \ldots, \lambda_c\}$$

The null hypothesis is defined as

$$H_{0,j} : R(\lambda_j) > \alpha, \qquad j \in \{1, \ldots, c\},$$

Notice that rejecting the null hypothesis corresponds to selecting λ_j as a point where the risk is controlled. For each null hypothesis, we compute a valid p-value using a concentration inequality such as Hoeffding–Bentkus (Bates et al., 2021). Family-wise error rate (FWER) control is then enforced across all candidates using either Bonferroni correction or Sequential Graphical Testing (SGT) (Bretz, Maurer, Brannath, & Posch, 2009), yielding the admissible set

$$\Lambda_{\text{adm}} \subseteq \{\lambda_1, \ldots, \lambda_c\}$$

of parameters satisfying the risk constraint with confidence $1-\delta$. A final λ is selected from Λ_{adm} according to an application-dependent logic:

- $l_{\min}$ / $l_{\max}$: select the smallest / largest $\lambda \in \Lambda_{\text{adm}}$;
- $r_{\min}$ / $r_{\max}$: select the $\lambda \in \Lambda_{\text{adm}}$ minimising / maximising the empirical risk $\widehat{R}_n(\lambda)$.

OOD detection. The first stage aims to detect trajectories inconsistent with the training distribution. For trajectory i, let $\widetilde{\mathbf{x}}_i^{(m)}$ denote its mth PCA-reduced segment, with $m \in \{1, \ldots, M_i\}$. To satisfy the i.i.d. assumption of LTT, calibration is performed at the trajectory level. The isolation forest assigns a segment-level OOD score s_{OOD} (cf. Eq. (6)), and the trajectory-level decision input is the largest segment score:

$$Z_i^{\text{OOD}} = \max_{m \in \{1,\ldots,M_i\}} s_{\text{OOD}}\left(\widetilde{\mathbf{x}}_i^{(m)}\right),$$

which captures the strongest evidence of distribution shift within trajectory i. The calibration dataset is formed from a validation set of trajectories with known in-distribution status:

$$\mathcal{D}_{\text{cal}}^{\text{OOD}} = \left\{\left(Z_i^{\text{OOD}}, Y_i\right)\right\}_{i=1}^{n_{\text{traj}}},$$

where $Y_i \in \{0, 1\}$ indicates whether trajectory i is out-of-distribution and n_{traj} is the number of trajectories in the calibration set. The threshold-based decision rule

$$\mathcal{S}_\lambda\big(Z^{\text{OOD}}\big) \in \{0, 1\}$$

is calibrated on $\mathcal{D}_{\text{cal}}^{\text{OOD}}$ to control the false alarm rate

$$\text{FAR}(\lambda) = \mathbb{P}\big(\mathcal{S}_\lambda(Z^{\text{OOD}}) = 1 \,\big|\, \mathbf{X} \text{ in-distribution}\big) \leq \alpha_{\text{OOD}}$$

with confidence $1 - \delta$. Trajectories flagged as OOD are excluded from all subsequent stages, ensuring that downstream risk guarantees are enforced conditionally on in-distribution data.

Fault detection. For in-distribution trajectories, fault detection determines whether an abnormal parametric change is present. As in OOD rejection, calibration is performed at the trajectory level to satisfy the i.i.d. assumption of LTT. Using the same segment notation, the classifier produces a segment-level detection score p_{det} (cf. Eq. 5). The trajectory-level decision input is the largest segment score:

$$Z_i^{\text{det}} = \max_{m \in \{1,\ldots,M_i\}} p_{\text{det}}\left(\mathbf{x}_i^{(m)}\right),$$

which captures the strongest evidence of a fault within trajectory i. The calibration dataset is formed from in-distribution trajectories, excluding those flagged as OOD:

$$\mathcal{D}_{\text{cal}}^{\text{det}} = \left\{\left(Z_i^{\text{det}}, Y_i\right)\right\}_{i=1}^{n_{\text{traj}}},$$

where $Y_i \in \{0, 1\}$ indicates whether trajectory i contains a fault. The threshold-based rule

$$\mathcal{S}_\lambda\big(Z^{\text{det}}\big) \in \{0, 1\}$$

is calibrated on $\mathcal{D}_{\text{cal}}^{\text{det}}$. Because false positives are particularly costly in our application, the primary controlled risk is the precision of the detector, with the FPR as an alternative formulation of the same objective. Recall is treated as a secondary risk to ensure that the controller does not overlook an excessive number of faults. We control the following risks/metrics:

$$P(\lambda) = \mathbb{P}\big(Y = 1 \,\big|\, \mathcal{S}_\lambda(Z^{\text{det}}) = 1\big),$$

$$\text{FPR}(\lambda) = \mathbb{P}\big(\mathcal{S}_\lambda(Z^{\text{det}}) = 1 \,\big|\, Y = 0\big),$$

$$R(\lambda) = \mathbb{P}\big(\mathcal{S}_\lambda(Z^{\text{det}}) = 1 \,\big|\, Y = 1\big).$$

Multiclass diagnosis with prediction sets. When a fault is detected, the diagnosis stage assigns a *prediction set* of plausible classes rather than a single label. For each segment $\mathbf{x}^{(m)}$, the classifier produces a probability vector $\hat{\mathbf{p}}(\mathbf{x}^{(m)}) = (\hat{p}_0(\mathbf{x}^{(m)}), \ldots, \hat{p}_K(\mathbf{x}^{(m)}))$ (cf. Section 3.3).

The segment-level decision input is

$$Z_m^{\text{diag}} = \hat{\mathbf{p}}(\mathbf{x}^{(m)}),$$

and the decision rule is

$$\mathcal{S}_\lambda\big(Z^{\text{diag}}\big) \subseteq \{0, \ldots, K\},$$

so all classes whose predicted probability meets or exceeds the threshold λ are included in the prediction set.

Calibration is performed by pooling eligible faulty segments across trajectories, namely segments whose trajectory is labelled as faulty ($Y \neq 0$) and that are predicted as faulty. Denoting by $\mathcal{M}$ the retained segment indices, the calibration

dataset is

$$\mathcal{D}_{\text{cal}}^{\text{diag}} = \left\{ \left(Z_m^{\text{diag}},\, y^{(m)} \right) \right\}_{m \in \mathcal{M}},$$

where $y^{(m)}$ is the true fault class of segment m. The controller is fitted on $\mathcal{D}_{\text{cal}}^{\text{diag}}$ to determine λ such that coverage and false discovery rate (FDR) satisfy user-defined targets with finite-sample confidence $1 - \delta$:

$$C(\lambda) = \mathbb{P}\big(Y \in \mathcal{S}_\lambda(Z^{\text{diag}})\big)\,,$$

$$\text{FDR}(\lambda) = \mathbb{E}\left[\frac{\left| \mathcal{S}_\lambda(Z^{\text{diag}}) \setminus \{Y\} \right|}{\left| \mathcal{S}_\lambda(Z^{\text{diag}}) \right|} \;\middle|\; \mathcal{S}_\lambda(Z^{\text{diag}}) \neq \emptyset \right].$$

Unlike the two preceding stages, no trajectory-level aggregation is applied here. While scalar quantities such as s_{OOD} and p_{det} naturally admit max-aggregation across segments, the multiclass probability vector $\hat{\mathbf{p}}$ does not have a canonical trajectory-level reduction. Alternatives such as averaging probabilities or taking per-class maxima across segments were evaluated but consistently produced inferior results. A likely explanation is that trajectory-level aggregation reduces the effective calibration set size, which is already smaller than in the previous stages by construction. Segment-level calibration instead pools all eligible fault segments across trajectories, yielding a larger calibration set. This comes at the cost of only approximately satisfying the i.i.d. assumption underlying LTT, as segments originating from the same trajectory may exhibit dependence.

4. EXPERIMENTAL SETUP

4.1. Datasets

To enable a direct comparison with our previous method (Basora et al., 2025), we conducted the experiments on the same two simulated datasets: a development dataset (*dev*) used for model training and testing, and an independent validation dataset (*test2*) used for final evaluation.

The *dev* dataset contains 2696 trajectories (14,054 segments). It is split at the trajectory level into training, validation, and test subsets using a 60:20:20 ratio. The training split is used to fit the models. The validation split is used for both hyperparameter tuning and calibration of the decision thresholds in the LTT framework. The test split is used only for in-development evaluation.

The splits in *dev* are approximately class-balanced in the original multiclass setting, with each class (the nominal class and each individual fault class) representing approximately 11% of trajectories and segments.

The *test2* dataset contains 1995 trajectories (5877 segments) and has a different class distribution, with a higher proportion of nominal data. Specifically, the nominal class represents approximately 15% of trajectories. Because nominal trajectories are longer on average in *test2*, the nominal prevalence increases to approximately 26% at the segment level.

The class-imbalance issue arises only in the binary fault-detection setting, where all fault classes are merged into a single positive class. Under this formulation, the nominal class becomes the minority class in both datasets, particularly in *dev*, despite the approximately balanced multiclass distribution.

Fault classes 2 and 32 were excluded from the experimental evaluation because their substantial overlap with the nominal class prevents a satisfactory precision–recall trade-off under the risk-control constraints of the proposed framework. A detailed analysis of class separability is provided in Section 6.1.

To evaluate OOD detection, we generate three synthetic classes (9 trajectories each) by transforming training trajectories. For a time series x_i, we define the transformed sample as $x_i' = x_i \cdot \text{var} + \text{shift} + i \cdot \text{trend}$. The parameter settings are class 20 $(-1, 5, 0)$, class 21 $(1, 5, 0)$, and class 22 $(1, 5, 10^{-5})$, where each tuple gives $(\text{var}, \text{shift}, \text{trend})$. These synthetic trajectories are added to the *test* and *test2* splits.

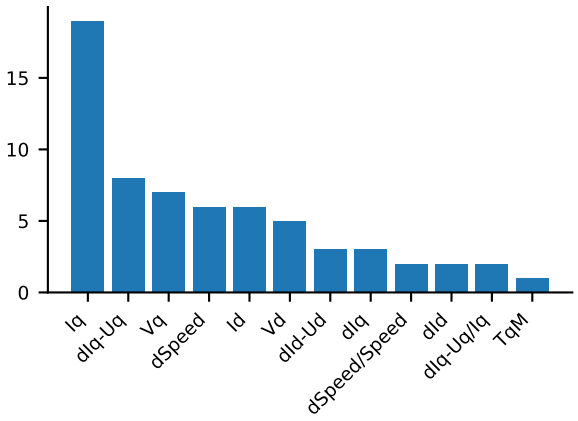
4.2. Pipeline configuration

Preprocessing. For signal segmentation, the minimum segment length and maximum gap were set to $L_{\min} = 50$ and $G_{\max} = 100$, respectively, to extract segments from trajectories. Visual inspection and statistical analysis confirmed that resulting segments correspond to periods of valve motion rather than spurious fluctuations or steady-state phases.

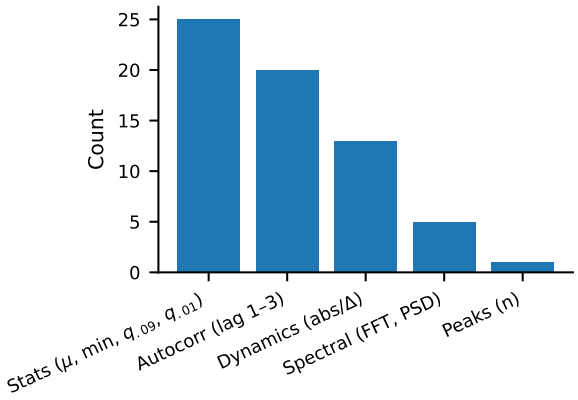
For feature selection, the number of retained features k was determined through a grid search over $k \in \{32, 64, 128, 256\}$. Among these configurations, $k = 64$ achieved the best trade-off between predictive performance and feature dimensionality. The selected `tsfresh` feature set comprises descriptors extracted from measured and reference electrical signals, mechanical variables, and derived error signals. These features include time-domain statistics, short-lag autocorrelation measures, change-based metrics, peak-related statistics, and a limited set of frequency-domain descriptors (see Figure 3).

Scoring models. The `LightGBM` hyperparameters were optimized using `FLAML` under a fixed budget of 100 trials, with the validation log-loss as the objective. The final `LightGBM` model was retrained using the selected configuration, which included the following key hyperparameters: *learning_rate* $= 0.1$, *n_estimators* $= 305$, *num_leaves* $= 10$, *min_child_samples* $= 128$, and regularization parameters $\ell_1 = 0.001$ and $\ell_2 = 0.046$.

The Isolation Forest was implemented using the default configuration provided by `scikit-learn`. The ensemble consists



(a) Selected features per signal.



(b) Distribution by feature family.

Figure 3. Structural summary of the $k = 64$ selected `tsfresh` features. The top panel shows the number of retained descriptors per signal, while the bottom panel groups features by functional family.

of 100 isolation trees, each trained on a random subsample of 256 observations. All PCA-derived features are provided to each base estimator. Segment-level feature vectors are projected onto a PCA subspace retaining at least 95% of the empirical variance, resulting in 11 principal components with a cumulative explained variance of 0.953.

Risk control. Decision thresholds were calibrated on the validation split using the LTT framework as specified in Section 3.5. We use the `mlrisko` library (Cordier, 2025), which provides a flexible implementation of LTT that natively supports multi-risk settings.

P-values were computed using the Hoeffding–Bentkus (HB) inequality, and FWER control was enforced via the SGT procedure, which empirically outperformed the Bonferroni correction. For fault detection, risk control enables an explicit precision–recall trade-off; for diagnosis, we control coverage and false discovery rate. From the admissible λ values returned by the procedure, the operational threshold was selected using the $l_{\min}$ rule for OOD detection and the $r_{\min}$ rule for fault detection and diagnosis, minimizing risks of low precision and low coverage, respectively. Experimental parameters are summarized in Table 4.

Table 4. Risk-control parameters by stage.

Stage	Parameter	Choice
General	δ (error rate)	0.1
General	p-value method	HB
General	FWER control	SGT
OOD	α_{OOD} (false alarm target)	0.1
OOD	λ choice rule	$l_{\min}$
OOD	c_{OOD} (number of thresholds)	200
FD	α_{prec} (precision target)	0.9
FD	α_{rec} (recall target)	0.7
FD	λ choice rule	$r_{\min}$
FD	$[\lambda_{\min}, \lambda_{\max}]$ (range)	[0.0, 1.0]
FD	c_{clf} (number of thresholds)	200
Diag.	α_{cov} (coverage target)	0.9
Diag.	α_{FDR} (FDR target)	0.2
Diag.	λ choice rule	$r_{\min}$
Diag.	$[\lambda_{\min}, \lambda_{\max}]$ (range)	[0.0, 1.0]
Diag.	c_{diag} (number of thresholds)	200

5. RESULTS

All reported performance metrics are computed over 10 independent runs with different random seeds and reported as mean $\pm$ standard deviation. The metrics are computed at the trajectory level.

5.1. OOD detection

Table 5 reports precision, recall, F1-score, and false alarm rate (FAR) for OOD detection on the *test* and *test2* splits. Performance is stable across datasets, with high precision and recall yielding F1-scores around 0.9. Importantly, FAR consistently remains well below the prescribed target ($\alpha_{\text{FA}} = 0.1$). The results on *test2* are similar despite its distribution shift, which indicates that the OOD detector remains robust under moderate prevalence changes. Variability across runs is small, particularly for precision and FAR, suggesting reliable operating behavior.

5.2. Fault detection

Table 6 summarizes fault detection performance on trajectories that pass the OOD filtering stage. Eligibility is high on both splits, indicating that most samples proceed to this stage. Precision remains stable at approximately 0.94, exceeding the prescribed target ($\alpha_{\text{prec}} = 0.9$), while the false positive rate stays low. Recall satisfies its minimum requirement ($\alpha_{\text{rec}} = 0.7$) on both datasets, although it decreases on *test2*, in line with an increase in the false negative rate. This

Table 5. OOD detection performance on the *test* and *test2* splits. Precision, recall, and F1-score are computed for the binary OOD-versus-in-distribution decision, and FAR denotes the false alarm rate. Each entry is reported as mean ± standard deviation over 10 runs with different random seeds. Target: $\alpha_{\mathrm{FA}} = 0.1$.

ds	test	test2
Precision	0.94 ± 0.02	0.96 ± 0.02
Recall	0.89 ± 0.17	0.89 ± 0.17
F1	0.91 ± 0.11	0.92 ± 0.11
FAR	0.05 ± 0.01	0.04 ± 0.01

suggests that the distribution shift in *test2* primarily affects missed detections rather than false alarms. The overall precision–recall trade-off remains within the calibrated operating region, as reflected by consistently high F1-scores.

Table 6. Fault detection performance on the *test* and *test2* splits for in-distribution trajectories. Eligibility denotes the fraction of trajectories not rejected as OOD. Precision, recall, F1-score, false positive rate (FPR), and false negative rate (FNR) are computed for the binary nominal-versus-fault decision. Each entry is reported as mean ± standard deviation over 10 runs with different random seeds. Targets: $\alpha_{\mathrm{rec}} = 0.7$, $\alpha_{\mathrm{prec}} = 0.9$.

ds	test	test2
Eligibility	0.91 ± 0.01	0.95 ± 0.01
Precision	0.94 ± 0.03	0.93 ± 0.03
Recall	0.80 ± 0.03	0.73 ± 0.02
F1	0.86 ± 0.02	0.82 ± 0.01
FPR	0.06 ± 0.03	0.06 ± 0.03
FNR	0.20 ± 0.03	0.27 ± 0.02

Table 7 reports the false negative rate (FNR), which varies substantially across classes. Faults 128 and 511 are perfectly detected on both splits, whereas classes such as 4 and 16 show moderate FNR. The particularly high FNR of classes 8, 64 and 256 reflects their intrinsic overlap with the nominal class (see Figure 5 and Section 6.1). FNR increases on *test2*, especially for these borderline classes, because even small shifts in the data distribution can move their scores below the fixed detection threshold, making them inherently more difficult to detect.

5.3. Diagnosis

Table 8 summarizes fault diagnosis performance on trajectories that pass the upstream OOD filtering and fault detection stages, resulting in a reduced set of eligible cases. Diagnosis is evaluated only on faulty trajectories correctly identified as faulty, since its objective is to determine the fault class (or set of plausible classes) once a fault has been detected. Among these eligible trajectories, coverage—the fraction of cases in which the true label is included in the prediction set—is very high (0.99 on both splits), exceeding the prescribed target

Table 7. False negative rate (FNR) per in-distribution fault class, computed after OOD filtering. For each fault label, FNR is the fraction of trajectories from that class that were incorrectly predicted as nominal by the binary fault detector. Each entry is reported as mean ± standard deviation over 10 runs with different random seeds.

ds	test	test2
ES=1	0.20 ± 0.08	0.24 ± 0.05
ES=4	0.10 ± 0.04	0.18 ± 0.04
ES=8	0.30 ± 0.07	0.39 ± 0.05
ES=16	0.12 ± 0.04	0.13 ± 0.01
ES=64	0.51 ± 0.07	0.69 ± 0.05
ES=128	0.00 ± 0.00	0.00 ± 0.00
ES=256	0.35 ± 0.05	0.53 ± 0.04
ES=511	0.00 ± 0.00	0.00 ± 0.00

($\alpha_{\mathrm{cov}} = 0.9$). The false discovery rate (FDR) remains below its target ($\alpha_{\mathrm{FDR}} = 0.2$), indicating that high coverage does not come at the expense of unreliable predictions. Precision and recall are quite stable across datasets, resulting in consistently high F1-scores. Overall, the diagnosis stage maintains its risk guarantees and predictive quality, with only minor variation under distribution shift.

Table 8. Fault diagnosis performance on the *test* and *test2* splits for faulty trajectories correctly identified as faulty. Eligibility denotes the fraction of trajectories reaching the diagnosis stage. Coverage is the proportion of cases in which the true fault label is contained in the prediction set, while FDR is the expected proportion of incorrect labels among all labels included in the prediction sets. Results are reported as mean ± standard deviation over 10 runs with different random seeds. Target risk levels: $\alpha_{\mathrm{cov}} = 0.9$ and $\alpha_{\mathrm{FDR}} = 0.2$.

ds	test	test2
Eligibility	0.68 ± 0.03	0.62 ± 0.02
Coverage	0.99 ± 0.01	0.99 ± 0.00
FDR	0.18 ± 0.03	0.18 ± 0.02
Precision	0.96 ± 0.02	0.96 ± 0.01
Recall	0.96 ± 0.02	0.96 ± 0.01
F1	0.95 ± 0.02	0.95 ± 0.01

The conformal inclusion matrix in Figure 4 extends the conventional confusion matrix to set-valued predictions, where entry (i, j) denotes the frequency with which label j appears in the prediction set when the true label is i. The matrix is computed on the combined *test* and *test2* splits for in-distribution samples detected as faulty after passing the two detection stages.

Most classes achieve near-perfect coverage, as reflected by the diagonal entries close to one. The main exception is class 0, which corresponds to false positives (from the fault detection stage) and frequently co-occurs with classes 64, 8, and 256. Among true fault classes, uncertainty is concentrated in a small subset of difficult faults, particularly 8, 64, and 1, whose prediction sets often include class 0 and other nearby faults.

Despite this ambiguity, coverage remains high, indicating that the conformal predictor captures uncertainty through larger prediction sets rather than incorrect predictions.

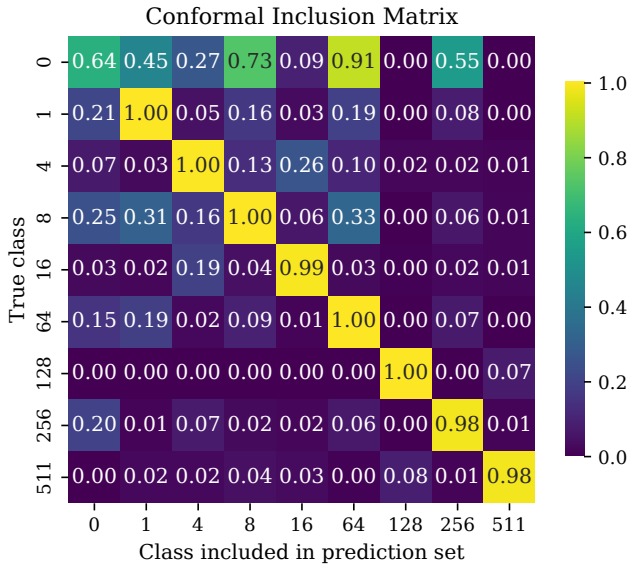


Figure 4. *test+test2* inclusion matrix for in-distribution detected fault cases. Rows correspond to the true labels and columns indicate the frequency with which each label appears in the conformal prediction set. Diagonal entries represent class coverage, while off-diagonal entries quantify inter-class uncertainty via label co-occurrence. The row for true class 0 corresponds to false positives from the fault detection stage.

This behavior is consistent with the PCA analysis (see 6.1), which identified these classes as the least separable faults and therefore the most likely to overlap with the nominal class or nearby faults.

6. DISCUSSION

6.1. Class separability

The PCA separability analysis (Figure 5) reveals that classes such as 2 and 32 exhibit significant overlap with nominal behavior. A large fraction of their samples falls inside the 95% nominal density region, resulting in very high nominal mass coverage. This is reflected in their small Bhattacharyya and Wasserstein distances: the Bhattacharyya distance measures how strongly two distributions overlap (smaller values mean more overlap), whereas the Wasserstein distance measures how far one distribution must be "shifted" to match the other in the embedding space (smaller values mean the distributions are closer). Low values for both metrics indicate that the faulty and nominal trajectories occupy nearly the same region in feature space, making them difficult to distinguish for any statistical classifier.

This PCA separability analysis is consistent with the conformal inclusion matrix (Figure 4), where "hard" faults such as 8, 64, and 256 exhibit comparatively low coverage and frequently include label 0 in their prediction sets. This behavior indicates partial overlap with the nominal regime.

Increasing the representation capacity does not resolve this limitation. The visualizations are computed with 64 features, but expanding the feature set (e.g., to 128 features) yields only marginal improvement. Across multiple runs and configuration settings, these classes consistently exhibit false negative rates above 60%, confirming that the ambiguity is structural.

6.2. Comparison with previous approach.

Our previous method (Basora et al., 2025) could only discriminate a subset of three fault classes $\{16, 128, 511\}$ against nominal data. For the same subset of faults and dataset configuration, the proposed method substantially improves the precision–recall trade-off and reduces FNR by an order of magnitude without sacrificing precision. Despite the higher detection rate, diagnosis performance remains comparable. Table 9 summarizes the comparison across both test splits. The proposed method still achieves overall stronger performance with the five additional fault classes (see Tables 6–8).

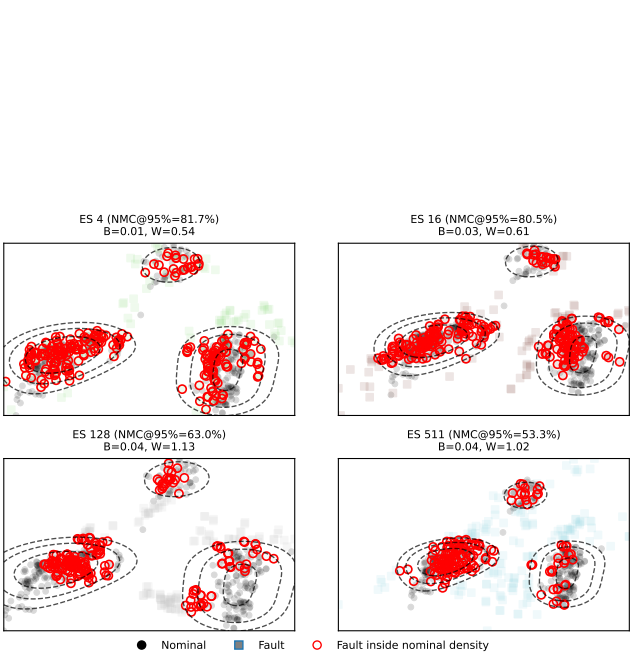
Table 9. Comparison of previous and current methods for fault detection (FD) and diagnosis (Diag) on the same fault subset. Each cell reports previous/current method metrics.

Task	Metric	test	test2
FD	Precision	0.95 / 0.97	0.91 / 0.98
	Recall	0.69 / 0.97	0.66 / 0.97
	F1	0.80 / 0.97	0.77 / 0.97
Diag	Precision	0.91 / 0.97	0.93 / 0.96
	Recall	0.93 / 0.97	0.94 / 0.96
	F1	0.92 / 0.96	0.93 / 0.95

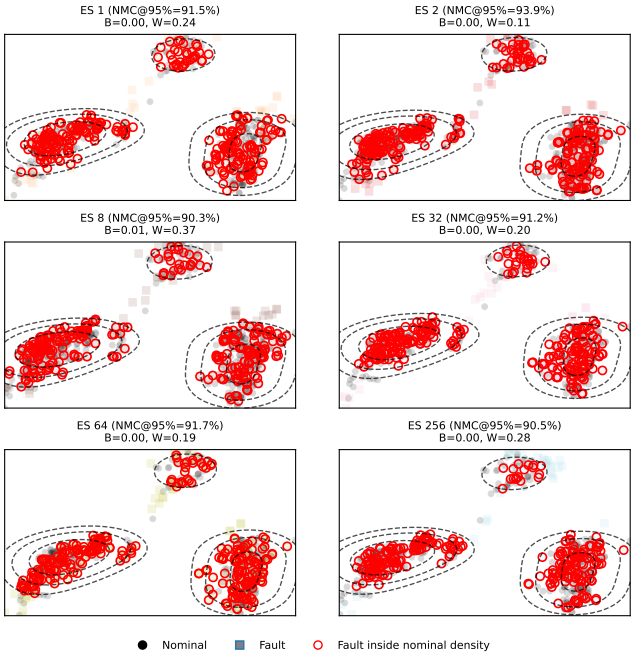
The performance gains are primarily due to the redesigned preprocessing and feature extraction pipeline. In the previous approach, the autoencoder operated on full trajectories using sliding windows over raw signals, including long, uninformative steady-state periods. In contrast, the proposed method leverages domain knowledge to isolate valve-motion intervals, where fault information is most pronounced.

A second key difference is the feature representation. Instead of relying solely on learned latent embeddings, the proposed pipeline extracts signal-specific features tailored to electromechanical systems, improving discriminative power. These features better capture transient dynamics, oscillations, timing irregularities, and fault-induced spectral variations, leading to a clearer separation between nominal and faulty trajectories, especially for more difficult fault classes.

Finally, the LTT calibration step in the fault detection stage selects operating thresholds with better sensitivity while controlling false positives, which also helps explain why the strongest improvements are observed in fault detection rather than fault



(a) "Easy" fault classes with NMC < 0.9.



(b) "Hard" fault classes with NMC ≥ 0.9.

Figure 5. PCA projections of nominal data versus all fault classes, split for readability. The division is based on nominal mass coverage (NMC): easy classes (left) have NMC < 0.9, while hard classes (right) have NMC ≥ 0.9. Dashed contours indicate the 95% nominal density, and fault samples falling inside this region are highlighted. Each panel reports Bhattacharyya distance (B), Wasserstein distance (W), and nominal mass coverage (NMC).

diagnosis.

6.3. Operational deployment

Prior to operational deployment, further validation on real-system data is required. An initial adaptation step involves recalibrating decision thresholds to account for differences in class prevalence. The proposed pipeline is developed using simulated data with balanced class distributions. Although test-bench facilities are expected to provide fault-injection capabilities that complement nominal operating conditions with representative faulty cases, nominal behavior is likely to dominate in the target environment, and fault prevalence may change over time.

If operational data differ substantially from the simulated training data due to sensor noise, unmodeled dynamics, or environmental variability, threshold recalibration alone may be insufficient, making partial or full model retraining necessary. Continuous monitoring of score distributions, OOD rejection rates, and prediction-set sizes can provide early indicators of distribution shift and help determine when model updates are required. In settings where labeled operational data remain limited, domain adaptation and transfer learning techniques offer promising alternatives and constitute a natural direction for future work.

7. CONCLUSIONS

This paper introduces a methodology for detecting and diagnosing abnormal parametric variations in electrically actuated valves for next-generation space launchers. The proposed pipeline combines segmentation of informative valve-motion intervals, automated feature extraction using `tsfresh`, and multiclass classification with `LightGBM`, with reliability enforced post-training through an LTT risk-control layer providing finite-sample guarantees for OOD rejection, fault detection, and diagnosis via prediction sets.

Experiments on simulated trajectories demonstrate stable performance and consistent satisfaction of prescribed risk targets on both the development test split and an independent validation dataset. In fault detection, precision is high and the FPR remains around 5%, limiting false positives, which are considered more costly than false negatives. The diagnosis stage achieves high coverage while keeping the FDR below the specified threshold, demonstrating reliable uncertainty quantification without overly conservative prediction sets.

A class-separability analysis reveals an intrinsic limitation of the available data for near-nominal fault modes. Classes 2 and 32 were excluded due to strong overlap with nominal behavior, and similar difficulties are observed for classes 64 and 256. Whether this reflects a fundamental limitation of the simulated

data remains an open question to be investigated with real data.

From a deployment perspective, the separation between score generation and LTT-based calibration provides a practical means to adapt decision thresholds to changing operational conditions, such as shifts in class prevalence. When recalibration is insufficient, retraining on representative operational data will be necessary.

Future work will focus on operational validation on real test-bench data, including the use of fault-injection capabilities to complement nominal observations. Further investigations should also address improved observability for strongly overlapping fault modes, and adaptive strategies for deployment under distribution shift, such as domain adaptation or transfer learning when labeled operational data are scarce.

The proposed methodology can be adapted to other PHM applications involving multivariate time-series data, with the caveat that risk control requires exchangeability and LTT requires i.i.d. data. These assumptions warrant careful attention in time-series settings where temporal dependencies may violate them.

ACKNOWLEDGMENT

This document was produced as part of the ENLIGHTEN-ED program, funded by Horizon Europe under Grant Agreement No. 101135156. The authors gratefully acknowledge the support of ArianeGroup for initiating the activity and providing the data used in the illustrations.

REFERENCES

Angelopoulos, A. N., Bates, S., Candès, E. J., Jordan, M. I., & Lei, L. (2025). Learn then test: Calibrating predictive algorithms to achieve risk control. *The Annals of Applied Statistics*, *19*(2), 1641–1662.

Bagnall, A., Lines, J., Bostrom, A., Large, J., & Keogh, E. (2017). The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data mining and knowledge discovery*, *31*, 606–660.

Basora, L., Bocquet-Nouaille, L., Robinson, E., & Gonidec, S. L. (2025). Fault detection and diagnosis for the engine electrical system of a space launcher based on a temporal convolutional autoencoder and calibrated classifiers. *arXiv preprint arXiv:2507.13022*.

Bates, S., Angelopoulos, A., Lei, L., Malik, J., & Jordan, M. (2021). Distribution-free, risk-controlling prediction sets. *Journal of the ACM (JACM)*, *68*(6), 1–34.

Benbouzid, M. E. H. (2000). A review of induction motors signature analysis as a medium for faults detection. *IEEE transactions on industrial electronics*, *47*(5), 984–993.

Bretz, F., Maurer, W., Brannath, W., & Posch, M. (2009). A graphical approach to sequentially rejective multiple test procedures. *Statistics in Medicine*, *28*(4), 586-604. doi: https://doi.org/10.1002/sim.3495

Choi, K., Yi, J., Park, C., & Yoon, S. (2021). Deep learning for anomaly detection in time-series data: Review, analysis, and guidelines. *IEEE access*, *9*, 120043–120065.

Christ, M., Braun, N., Neuffer, J., & Kempa-Liehr, A. W. (2018). Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package). *Neurocomputing*, *307*, 72–77.

Cordier, T. (2025). *Risk control for machine learning.* (Available at: thibaultcordier.github.io/risk-control)

Diallo, A. R., Homri, L., & Dantan, J.-Y. (2025). Reducing false alarms in fault detection: A comparative analysis between conformal prediction and classical methods applied to PCA and autoencoders. *Journal of Process Control*, *152*, 103495.

Férard, B., Le Gonidec, S., Galeotta, M., Oriol, S., & Dreyer, S. (2021). Anomaly detection on propulsive systems by global approach using autoencoders. *IFAC-PapersOnLine*, *54*(4), 31–37.

Heddoub, A., Diallo, A. R., Homri, L., Dantan, J.-Y., & Siadat, A. (2025). Uncertainty-Aware Fault Diagnosis with Conformal Prediction. *IFAC-PapersOnLine*, *59*(10), 536–541.

Husebø, A. B., Kandukuri, S. T., Klausen, A., Huynh, K., & Robbersmyr, K. G. (2020). Rapid diagnosis of induction motor electrical faults using convolutional autoencoder feature extraction. *Annual Conference of the PHM Society*.

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., & Muller, P.-A. (2019). Deep learning for time series classification: a review. *Data mining and knowledge discovery*, *33*(4), 917–963.

Javanmardi, A., & Hullermeier, E. (2023). Conformal Prediction Intervals for Remaining Useful Lifetime Estimation. *International Journal of Prognostics and Health Management*, *14*(2).

Jiménez-Guarneros, M., Morales-Perez, C., & de Jesus Rangel-Magdaleno, J. (2021). Diagnostic of combined mechanical and electrical faults in ASD-powered induction motor using MODWT and a lightweight 1-D CNN. *IEEE Transactions on Industrial Informatics*, *18*(7), 4688–4697.

Kayali, M., & Wang, C. (2022). Mining robust default configurations for resource-constrained AutoML. *arXiv preprint arXiv:2202.09927*.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, *30*.

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. In *2008 eighth ieee international conference on data mining* (pp. 413–422).

Nandi, S., Toliyat, H. A., & Li, X. (2005). Condition monitoring and fault diagnosis of electrical motors—a review. *IEEE transactions on energy conversion*, *20*(4), 719–729.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Robinson, C. (2026). Remaining Useful Life Estimation for Aircraft Engines with Risk-Aware Prediction Intervals via Conformalized Quantile Regression. *International Journal of Prognostics and Health Management*, *17*(1).

Shafer, G., & Vovk, V. (2008). A tutorial on conformal prediction. *Journal of Machine Learning Research*, *9*(3).

Shao, S., Yan, R., Lu, Y., Wang, P., & Gao, R. X. (2019). DCNN-based multi-signal induction motor fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, *69*(6), 2658–2669.

Thomson, W. T., & Fenger, M. (2003). Case histories of current signature analysis to detect faults in induction motor drives. In *Ieee international electric machines and drives conference, 2003. iemdc'03.* (Vol. 3, pp. 1459–1465).

Verma, S. (2023). *Deep neural network modeling of electric motors* (PhD thesis). Université Paris-Saclay. (HAL Id: tel-04231692)

Verma, S., Henwood, N., Castella, M., Malrait, F., & Pesquet, J.-C. (2020). Modeling electrical motor dynamics using encoder-decoder with recurrent skip connection. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 34, pp. 1387–1394).

Wang, C., Wu, Q., Weimer, M., & Zhu, E. (2021). Flaml: A fast and lightweight AutoML library. In *Mlsys.*

Yang, C.-L., Meles, T. Y., Yilma, A. A., & Teshome, M. M. (2026). Uncertainty aware predictive maintenance using a hybrid Transformer with Monte Carlo Dropout and conformal prediction. *Ain Shams Engineering Journal*, *17*(2), 103992.