Resource-Aware Model Comparison for DASHlink Multi-Class Flight-Anomaly Screening

Keanan Milton[1] and Ge Zhou[2]

[1, 2] Belcan Engineering, a Cognizant company, Windsor, CT, 06095, USA
kmilton@belcan.com
gzhou@belcan.com

ABSTRACT

Flight-data prognostics and health management models have to do more than rank anomalous windows correctly. For future edge or onboard validation, a screening model also has to be compact, fast to score, false-alarm aware, and interpretable enough that its decision statistic can be trusted. Autoencoders are attractive in this setting because they can be trained from nominal data, but a single scalar reconstruction error may be too coarse to separate operationally different non-nominal approach behaviors. We tested that tradeoff on the National Aeronautics and Space Administration (NASA) Discovery in Aeronautics Systems Health (DASHlink) multi-class flight-anomaly window benchmark using a fixed split, nominal-only threshold calibration, and a held-out test set. Each sample is a 160-second, 20-variable window labeled as Nominal, Speed High, Path High, or Flaps Late Setting. We treat the primary task as nominal-versus-non-nominal screening, while preserving class-wise recall so that the rare classes are not hidden by aggregate recall. The comparison includes raw-summary classifiers, classical unsupervised baselines, scalar convolutional variational autoencoder (CVAE) reconstruction scores, CVAE residual and latent representations, and raw-plus-CVAE hybrid models. Thresholded metrics use a nominal-calibrated p95 operating point; resource proxies include total scoring time and serialized pipeline size. Scalar CVAE reconstruction error was weak as a standalone detector, with AP 0.179 and worst-class recall 0.051. Residual and latent CVAE features retained substantially more class-discriminative information, and the strongest raw-plus-CVAE hybrid reached AP 0.935, recall 0.945, and worst-class recall 0.894 at 0.158 ms/window. Full-20 raw-summary Histogram Gradient Boosting (HGB) remained the simpler high-performing baseline. The practical result is that compact feature-based models are strong screening baselines, while reconstruction models are most useful here as auxiliary representation generators rather than standalone scalar anomaly detectors.

1. INTRODUCTION

In flight-data monitoring, a useful Prognostics and Health Management (PHM) screening model cannot be judged by ranking performance alone. A detector that is accurate but slow, large, poorly calibrated, or difficult to interpret may still be unattractive for an edge- or onboard-oriented workflow. This is why the evaluation in this paper treats false alarms, latency, model size, and evidence traceability as first-class quantities alongside detection performance (Vachtsevanos et al., 2006; Schwabacher & Goebel, 2007). The issue is especially sharp for reconstruction models. They are appealing when non-nominal examples are limited, but a single reconstruction score can collapse multichannel time-varying behavior into one number and miss class-specific signatures (Chandola et al., 2009).

Prior DASHlink flight-anomaly studies have emphasized reconstruction-based detection and fault disambiguation, semi-supervised multiclass learning, and, more recently, edge deployment of an autoencoder-based detector on Raspberry Pi-class hardware (Reddy et al., 2016; Memarzadeh et al., 2020, 2022; Dudukcu et al., 2025). Those studies establish that learned representations can be effective and that resource-constrained deployment is feasible, but they do not directly isolate the representation-versus-resource question addressed here. Using the public four-class DASHlink window artifact (National Aeronautics and Space Administration, n.d.), we compare compact raw summaries, classical unsupervised scores, scalar reconstruction error, residual/latent CVAE features, and raw-plus-CVAE hybrids under one nominal-calibrated held-out operating rule. The goal is not a new autoencoder architecture; it is to determine whether CVAE representation machinery improves screening enough to justify its added scoring cost and pipeline complexity.

The paper makes four contributions. First, it separates model selection, nominal-only threshold calibration, and held-out testing so that the reported operating point is not chosen from the test labels. Second, it evaluates each model with both detection metrics and resource proxies, including total Python scoring time and serialized artifact size. Third, it

directly separates scalar-score value from representation value by testing whether CVAE information is more useful as one reconstruction score or as residual/latent features. Fourth, it includes a stabilized CVAE run to check whether the weak scalar-CVAE result was merely an optimization artifact; that check did not change the final interpretation.

2. Dataset and Experimental Design

We used fixed-length multivariate windows from the public DASHlink curated four-class anomaly corpus, which has also been used for semi-supervised multiclass learning and edge-deployment studies (National Aeronautics and Space Administration, n.d.; Memarzadeh et al., 2022; Dudukcu et al., 2025). Each sample is a 160-second sequence with 20 time-series variables. For the main screening task, Speed High, Path High, and Flaps Late Setting are grouped as non-nominal, but we keep the original class labels for worst-class recall and supplemental four-class analysis. That choice keeps the operational question simple while still exposing failures on rare event types.

The processed .npz artifact provides the windows and labels used in all experiments, but it does not expose the original flight grouping, window-overlap policy, label-generation workflow, or tail-level identifiers. We therefore treat the file as a benchmark artifact rather than as an independently verified event-causality record. The analysis measures whether the documented labels are separable under a fixed protocol; it does not verify maintenance relevance, causal fault mechanisms, or aircraft-level generalization.

The labels are Nominal, Speed High, Path High, and Flaps Late Setting. Because flight identifier, tail identifier, maintenance confirmation, and true event onset time are unavailable in the final artifact, every claim is bounded to window-level screening. The split is controlled at the window and protocol level, but we cannot prove flight- or tail-level independence from the available metadata. This limitation is important when interpreting leakage risk and deployment relevance.

Table 1. Dataset class distribution.

Class	N	%
Nominal	89663	89.810
Speed High	7013	7.020
Path High	2207	2.210
Flaps Late	954	0.960

Note. Each sample is a 160-second window with 20 variables.

Table 2. Locked split counts.

Subset	Total	Nom.	Speed	Path	Flap
Held-Out	29,952	26,900	2,104	662	286
Validation	6,100	5,000	700	250	150
Nom. Cal.	10,000	10,000	0	0	0
Selection	53,785	47,763	4,209	1,295	518
Final Train	59,885	52,763	4,909	1,545	668

Note. Nom. Cal. uses nominal-only windows; test labels are not used for threshold selection. Splits were generated once with a fixed seed and stratified by class where labels were used.

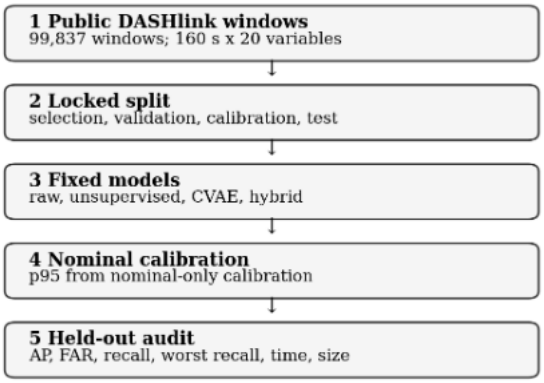

Figure 1. Locked experimental protocol.

3. Methods

3.1. Feature Sets and Model Families

We evaluated two channel policies. Full-20 uses all measured variables and is treated as the strongest empirical feature set. Primary-18 removes selected course and selected heading so that the measured-response baseline is less dependent on command/reference channels. For raw-summary models, each channel is reduced to compact window descriptors, including mean, standard deviation, extrema, selected percentiles, final value, range, and start-to-end change. This representation intentionally trades temporal detail for a small, implementation-friendly feature vector.

The model set includes logistic regression, ExtraTrees, Isolation Forest, principal-component reconstruction, Histogram Gradient Boosting (HGB), convolutional autoencoder (CAE), CVAE, residual/latent HGB models, and raw-plus-reconstruction hybrid HGB models. HGB is the main nonlinear reference because it models threshold-like interactions in tabular summaries without requiring a large neural decision layer (Friedman, 2001). Using the same HGB decision model for raw, residual, latent, and hybrid inputs also makes it easier to interpret the value of each representation.

The other baselines serve specific checks rather than proposed deployment roles. Logistic regression tests whether a linear summary model is sufficient. Principal Component Analysis (PCA) reconstruction and Isolation Forest test classical unsupervised scoring. ExtraTrees tests whether a larger tree ensemble changes the performance-cost frontier.

These baselines keep the comparison grounded; they are not treated as the final recommended monitor.

3.2. CAE/CVAE Design Rationale

The CAE/CVAE family is used as a lightweight reconstruction-feature baseline, not as a claimed architectural contribution. The CVAE follows the variational-autoencoder foundation of Kingma and Welling (2014), and we included it because prior flight-data studies have used autoencoder and convolutional variational autoencoder representations for anomaly detection and fault disambiguation (Reddy et al., 2016; Memarzadeh et al., 2020). Development-stage studies motivated the Primary-18 reconstruction input, no skip connections, temporal convolutional encoder/decoder, latent dimension 64, non-saturating activations, and low-beta objective.

In this study, scalar CVAE RMSE means that the full time-channel reconstruction mismatch is collapsed into one global score. Residual features instead summarize where the reconstruction was wrong by comparing each input window with its reconstruction. Latent features are the encoder's compressed internal representation of the window. This distinction matters because localized speed, path, or flap-related deviations can be diluted when all reconstruction errors are averaged into one scalar value.

We retained the no-skip design because skip paths can make reconstruction easier while reducing the contrast in residuals, which is exactly the evidence needed if residuals are used downstream. The low-beta CVAE setting was retained because earlier development runs favored reconstruction-feature utility over heavily regularized generative behavior. The stabilized CVAE run used a lower learning rate and gradient clipping to test a narrow concern: whether scalar-CVAE weakness was caused by unstable optimization. It preserved the same held-out protocol and did not tune thresholds on test labels, so we interpret it as a robustness check rather than a new post-hoc model-selection loop.

Table 3. Reconstruction-design settings.

Choice	Final setting
Input	Primary-18 response channels
Architecture	Conv1D / ConvTranspose1D
Skip paths	None
Latent dim.	64
CVAE beta	0.005
Stability patch	LR 3e-4; clip 1.0

Note. Rationale is described in the surrounding text; settings are not claimed to be globally optimal.

3.3. Operating Point and Resource Metrics

All thresholded scores are calibrated from nominal windows only. The main operating point is the 95th percentile of nominal calibration scores, denoted p95, which fixes the false-alarm operating point before the held-out test labels are evaluated. Average precision (AP) summarizes the ranked precision-recall curve and is emphasized because the non-nominal classes are imbalanced (Davis & Goadrich, 2006). At the p95 threshold, we report false-alarm rate (FAR), aggregate non-nominal recall, and worst non-nominal class recall. In classical detection terminology, FAR is the empirical estimate of false-alarm probability PFA on nominal held-out windows, while recall is the empirical estimate of detection probability PD for labeled non-nominal windows. These operating characteristics do not require a particular parametric detector. Bayesian classifiers and multiple-hypothesis tests are valid complementary formulations, but adding them would change the detector family rather than the representation comparison studied here; we therefore treat them as future baselines rather than implying that the available data preclude their use.

For a window x_i with T time steps and C channels, label y_i, score s_i, and nominal calibration set C_nom, the threshold and thresholded metrics are defined in Eqs. (1)-(5).

$$\tau_{95} = Q_{0.95}(\{s_i : y_i = 0,\ i \in C_{nom}\}) \quad (1)$$

$$\hat{P}_{FA} \equiv FAR = FP/(FP + TN) \quad (2)$$

$$\hat{P}_{D} \equiv Recall = TP/(TP + FN) \quad (3)$$

$$Recall_{worst} = \min_{c \in \{1,2,3\}} TP_c/(TP_c + FN_c) \quad (4)$$

$$s_i = \sqrt{[(1/(TC)) \sum_{t=1}^{T} \sum_{j=1}^{C} (x_{ij} - \hat{x}_{ij})^2]} \quad (5)$$

Here Q_0.95 is the empirical 95th percentile over the nominal calibration set C_nom. FP and TN are nominal-window counts, TP and FN are non-nominal-window counts, and worst-class recall is the minimum recall across Speed High, Path High, and Flaps Late Setting. Equations (2) and (3) make explicit that FAR and recall are empirical PFA and PD estimates at the selected threshold. The scalar CVAE score in Eq. (5) is a global RMSE over time and channels; this deliberately tests the common assumption that one reconstruction statistic is enough for anomaly screening.

Resource measurements are reported as workstation proxies, not certified embedded runtimes. They include feature extraction time, model inference time, total Python scoring time per window, samples per second, classifier artifact size, autoencoder checkpoint size when applicable, and full pipeline artifact size. Hybrid timing includes raw feature extraction, CVAE forward pass, residual/latent construction, and HGB inference. The measurements were collected on Windows 10 with Python 3.11.11, PyTorch 2.5.1, CUDA 12.1, an NVIDIA RTX 4080 SUPER with 16 GB GPU memory, 16 physical/32 logical CPU cores, approximately 63 GB RAM, and seven repeated batch scoring passes.

The source package records the preprocessing assumptions, fixed random seed, model hyperparameters, scoring scripts, timing protocol, and artifact-size calculations used to generate the reported tables and figures. In the final stabilized reconstruction run, CVAE training used nominal-only windows, latent dimension 64, beta 0.005, learning rate 3e-4, batch size 128, gradient clipping at 1.0, an 80-epoch cap, and

best-validation checkpoint selection. These details are included to make the calculation pipeline auditable; they should not be read as a claim that the neural configuration is globally optimal.

4. RESULTS

4.1. Primary Held-out Benchmark

Table 4 gives the primary held-out comparison. Full-20 raw HGB is the strongest low-cost baseline: AP 0.916, worst-class recall 0.872, 0.076 ms/window, and 1.25 MB. The Full-20+CVAE hybrid gives the strongest observed detection performance: AP 0.935, non-nominal recall 0.945, and worst-class recall 0.894 at 0.158 ms/window and 3.38 MB. That gain is useful but not free. We interpret the hybrid as an accuracy-cost tradeoff, while raw HGB remains the simpler deployment-oriented baseline.

The scalar CVAE result is the clearest negative result in the paper. Scalar CVAE RMSE reached only AP 0.179 and worst-class recall 0.051, which is too weak for a standalone monitor. By contrast, the CVAE residual+latent model reached AP 0.853 and worst-class recall 0.833. The difference indicates that the reconstruction model does contain useful information, but that information is distributed across residual patterns and latent coordinates rather than captured by one global RMSE. Worst-class recall is the key guardrail here because it exposes rare-class failure that aggregate recall can hide.

Table 4. Main held-out comparison at nominal p95.

Model	AP	FAR	Rec.	Worst	ms	MB
F20+CVAE hybrid	0.935	0.049	0.945	0.894	0.158	3.380
P18+CVAE hybrid	0.933	0.051	0.945	0.890	0.151	3.341
F20 raw HGB	0.916	0.051	0.926	0.872	0.076	1.246
P18 raw HGB	0.912	0.051	0.919	0.843	0.069	1.208
CVAE res+lat	0.853	0.051	0.854	0.833	0.084	2.988
CVAE scalar	0.179	0.050	0.152	0.051	0.080	1.723

Note. AP is threshold-independent. At the nominal p95 threshold, FAR is the empirical PFA on nominal held-out windows and Rec. is the empirical PD on labeled non-nominal windows; Worst is the minimum class-wise recall across the three non-nominal classes.

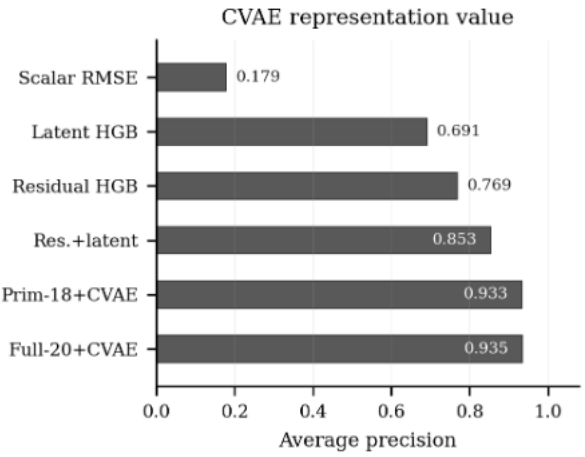

Figure 2. CVAE scalar error is weak; residual/latent and hybrid features are stronger.

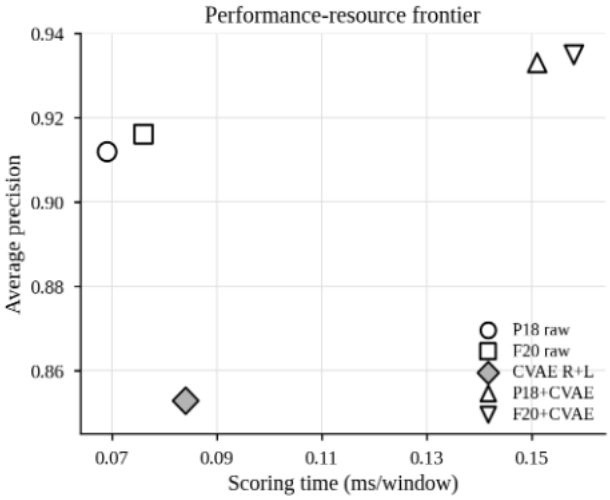

Figure 3. Performance-resource frontier. Marker shapes identify the model families; scalar CVAE RMSE is shown separately in Figure 2.

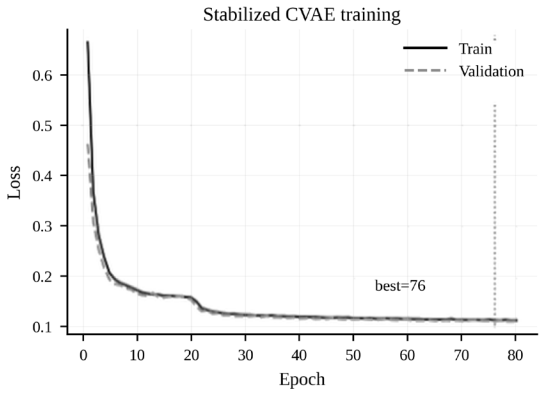

Figure 4. Stabilized CVAE training curve with lower learning rate and gradient clipping.

4.2. Threshold Stability and Uncertainty

We also checked whether the conclusion depends on one favorable threshold. Table 5 evaluates nominal calibration percentiles p90, p95, p97.5, and p99. As expected, stricter thresholds reduce false alarms and increase precision while lowering recall. Across these settings, scalar CVAE RMSE remains weak, whereas the Full-20+CVAE hybrid retains high recall over a broad operating range.

Bootstrap intervals provide a second check on the main interpretation (Efron & Tibshirani, 1993). We used non-stratified percentile-bootstrap intervals from 1000 resamples of held-out windows, with the nominal calibration scores held fixed. In each resample, held-out windows are sampled with replacement, the metrics are recomputed, and the 2.5th and 97.5th percentiles define the reported 95 percent interval. These intervals describe sampling uncertainty for the held-out window set. They do not capture uncertainty from alternative flight-level splits, because flight and tail identifiers are not available in the artifact.

Table 5. Operating-point stability examples.

Model	pctl	FAR	Prec.	Rec.	Worst
CVAE RMSE	95.00	0.050	0.255	0.152	0.051
CVAE RMSE	99.00	0.010	0.204	0.023	0.005
Full-20+CVAE	95.00	0.049	0.686	0.945	0.894
Full-20+CVAE	99.00	0.011	0.891	0.822	0.755

Note. Percentiles are computed on nominal calibration scores only.

Table 6. Bootstrap 95 percent confidence intervals.

Model	AP CI	Rec. CI	Worst CI
CVAE scalar	0.170-0.189	0.139-0.164	0.042-0.060
F20 raw HGB	0.908-0.924	0.916-0.935	0.845-0.896
CVAE res+lat	0.843-0.863	0.842-0.866	0.817-0.847
F20+CVAE hyb.	0.929-0.942	0.937-0.953	0.871-0.917

Note. AP CI is threshold-independent; Rec. CI and Worst CI are evaluated at the nominal p95 threshold on held-out windows.

4.3. Temporal Horizon and Class Behavior

The temporal-horizon ablation asks how much signal remains if the model sees less of the 160-second window. The 40-second horizon is fast and compact but loses substantial sensitivity. The 80-second horizon recovers much of the 160-second performance, while the full window remains strongest. This pattern is consistent with approach-window signatures that need enough time for speed, path, and flap-state behavior to separate from nominal variation.

The supplemental four-class HGB model confirms that the raw summaries preserve class-specific signal. Recall is high for all four classes, but precision is lower for Speed High and Path High, indicating off-diagonal confusion between non-nominal behaviors. That is why the main operational formulation remains nominal-versus-non-nominal screening, with class-wise metrics reported as a check against rare-class failure.

Table 7. Temporal horizon sensitivity.

s	AP	Rec.	Worst	ms
40.00	0.756	0.753	0.663	0.029
80.00	0.899	0.914	0.813	0.050
160	0.890	0.908	0.861	0.088

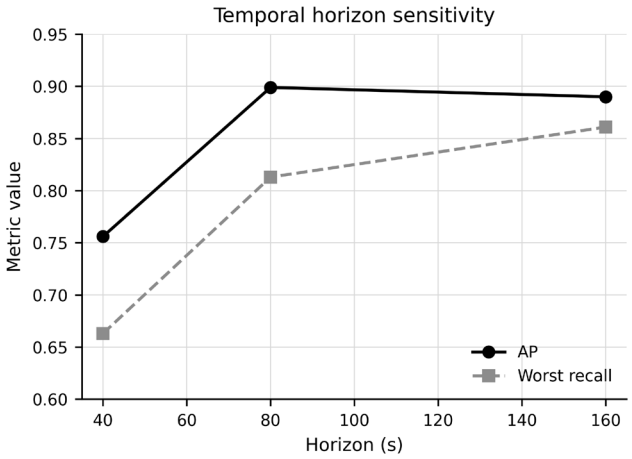

Figure 5. Shorter temporal horizons reduce sensitivity, especially at 40 seconds.

Table 8. Supplemental four-class HGB performance.

Class	N	Prec.	Rec.	F1
Nominal	26900	0.974	0.987	0.980
Speed High	2104	0.835	0.750	0.790
Path High	662	0.833	0.699	0.760
Flaps Late	286	0.934	0.839	0.884

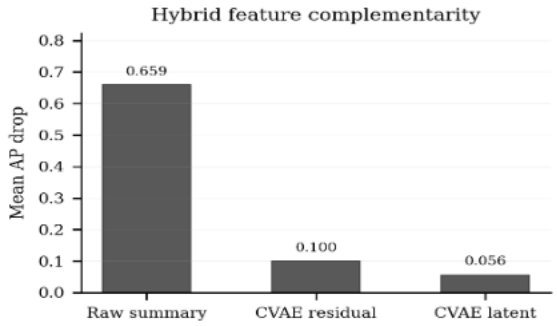

Figure 6. Feature-group complementarity. Raw-summary features dominate; CVAE residual and latent groups add nonzero AP signal.

5. DISCUSSION

The main lesson is that scalar reconstruction error is not a reliable decision statistic for this DASHlink task. The stabilized CVAE run makes that result harder to dismiss as an optimization failure: even after the stability patch, scalar RMSE produced poor AP and very low worst-class recall. At the same time, the CVAE did not fail to learn useful structure. Residual and latent features were far more informative, and hybrid HGB models improved AP and worst-class recall relative to raw HGB, albeit with higher pipeline cost. The evidence therefore points to representation value, not scalar-score value.

The deployment tradeoff is explicit. Full-20 raw HGB is the cleanest low-cost baseline and is easier to maintain because it uses only raw-summary features and a compact tree model. Full-20+CVAE hybrid HGB provides the best observed detection performance, but it requires an upstream neural reconstruction model and roughly doubles workstation-proxy scoring time relative to raw HGB. A constrained implementation would likely start with raw HGB; a less constrained edge setting could justify the hybrid if the additional sensitivity is worth the added complexity. Recent AnoSense work demonstrates that the same four-class DASHlink problem can be deployed on Raspberry Pi hardware after model pruning (Dudukcu et al., 2025). Our resource results serve a different purpose: they expose the relative pipeline cost of representation choices and should not be interpreted as embedded-hardware validation.

Published DASHlink results provide important external context but are not score-to-score benchmarks for Table 4 because their experimental protocols differ. Memarzadeh et al. (2022) targets semi-supervised multiclass learning, whereas Dudukcu et al. (2025) collapses the same four-class corpus to binary detection and evaluates a pruned autoencoder-based network on Raspberry Pi hardware. The present study instead locks a nominal-only calibration set and held-out window test to isolate representation value and pipeline cost. Table 4 is therefore an internally controlled benchmark; the published studies establish methodological and deployment context rather than directly comparable numerical baselines.

The development-stage ablations should be read in that context. They justify a representative reconstruction design for this comparison; they do not prove global optimality. The final claims rest on the fixed held-out protocol, nominal-only calibration, and consistent resource accounting. Keeping those roles separate helps avoid an overly optimistic interpretation of a dataset that does not expose flight-level grouping or event-causality metadata.

The comparison also clarifies what was not tested. Bayesian classifiers and multiple-hypothesis detectors would be useful complementary baselines because they can make probabilistic decision assumptions explicit and characterize PFA/PD tradeoffs; physics-informed AI could be valuable where governing relationships or operational constraints are available. We did not add these detector families post hoc because the present study is designed to isolate representation choice under one fixed nominal-calibrated protocol, not to claim exhaustive detector optimality. The empirical PFA and PD quantities reported here provide a common operating-point view for the models that were tested.

6. LIMITATIONS

The main limitation is the benchmark artifact itself. It does not provide flight identifiers, tail identifiers, maintenance confirmation, operational context, verified fault onset time, or the original label-generation workflow. The split is therefore controlled at the window-index and protocol level but cannot prove flight- or tail-level independence. The results should be interpreted as held-out window-screening evidence, not as certified fleet-level deployment performance or independently verified fault causality.

The resource results are also limited. Timing was measured as batch Python workstation-proxy scoring on the hardware described in Section 3.3. Embedded runtime, memory allocation behavior, fixed-point implementation, real-time scheduling, data bus integration, thermal constraints, and certification constraints were not measured. The resource numbers are useful for relative comparison, but they are not a substitute for target-hardware validation.

Finally, the CAE/CVAE settings are rational and traceable but not globally optimized. The reconstruction study tests whether scalar, residual, latent, and hybrid representations are useful under a common protocol; it does not claim that broader neural architecture search, Bayesian detection, multiple-hypothesis testing, or physics-informed modeling would not improve results. Direct numerical comparison to prior DASHlink studies is also limited because label formulation, windowing, supervision, preprocessing, feature policy, split protocol, and hardware measurement can differ. The cited DASHlink studies are therefore used for methodological context rather than score-to-score ranking.

7. CONCLUSION

We evaluated lightweight anomaly-screening models for DASHlink multi-class flight-anomaly windows using a fixed split, nominal-only threshold calibration, held-out testing, and resource-aware pipeline accounting. The stabilized CVAE run addressed the main training-stability concern and did not overturn the central result: scalar CVAE reconstruction RMSE remains weak, while residual/latent features and raw-plus-CVAE hybrids are much more informative.

Full-20 raw HGB is the simplest high-performing low-cost baseline, with AP 0.916 at 0.076 ms/window and 1.25 MB. Full-20+CVAE hybrid HGB gives the strongest observed detection performance, with AP 0.935, recall 0.945, and

worst-class recall 0.894 at 0.158 ms/window and 3.38 MB. The practical conclusion is deliberately narrow: reconstruction models are valuable here as auxiliary representation generators, but scalar reconstruction error alone is not a reliable monitoring statistic for this benchmark. More broadly for PHM screening, the result shows why representation value and decision-statistic value should be evaluated separately and why any sensitivity gain should be weighed against measurable latency and pipeline complexity.

NOMENCLATURE

AP	average precision
CAE	convolutional autoencoder
CI	confidence interval
$CVAE$	convolutional variational autoencoder
FAR	false-alarm rate
HGB	histogram gradient boosting
$RMSE$	root mean squared error
P_{FA}	empirical false-alarm probability
P_D	empirical detection probability
C	number of channels
T	number of time steps
x_i	input window
$\hat{x}_i$	reconstructed window
y_i	class label
s_i	model score
τ_{95}	nominal p95 decision threshold
$Q_{0.95}$	empirical 95th percentile operator
C_{nom}	nominal calibration set

REFERENCES

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, 41(3), Article 15. https://doi.org/10.1145/1541880.1541882

Davis, J., & Goadrich, M. (2006). The relationship between precision-recall and ROC curves. *Proceedings of the 23rd International Conference on Machine Learning*, 233-240. https://doi.org/10.1145/1143844.1143874

Dudukcu, H. V., Taşkıran, M., & Kahraman, N. (2025). AnoSense: Edge computing for real-time flight anomaly detection by using embedded deep neural networks. *Gümüşhane Üniversitesi Fen Bilimleri Dergisi*, 15(3), 797-808. https://doi.org/10.17714/gumusfenbil.1676270

Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. Chapman & Hall.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5), 1189-1232.

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational Bayes. *International Conference on Learning Representations*.

Memarzadeh, M., Matthews, B., & Avrekh, I. (2020). Unsupervised anomaly detection in flight data using convolutional variational auto-encoder. *Aerospace*, 7(8), 115. https://doi.org/10.3390/aerospace7080115

Memarzadeh, M., Matthews, B., & Templin, T. (2022). Multiclass anomaly detection in flight data using semi-supervised explainable deep learning model. *Journal of Aerospace Info rmation Systems*, 19(2), 83-97. https://doi.org/10.2514/1.I010959

National Aeronautics and Space Administration. (n.d.). Sample Flight Data - DASHlink. NASA DASHlink Collaborative Sharing Network. Retrieved June 18, 2026, from https://c3.ndc.nasa.gov/dashlink/projects/85/

Reddy, K. K., Sarkar, S., Venugopalan, V., & Giering, M. (2016). Anomaly detection and fault disambiguation in large flight data: A multi-modal deep auto-encoder approach. *Annual Conference of the Prognostics and Health Management Society*.

Schwabacher, M., & Goebel, K. F. (2007). A survey of artificial intelligence for prognostics. *Proceedings of the AAAI Fall Symposium*, Arlington, VA.

Vachtsevanos, G., Lewis, F. L., Roemer, M., Hess, A., & Wu, B. (2006). *Intelligent fault diagnosis and prognosis for engineering systems*. John Wiley & Sons.

BIOGRAPHIES

Keanan Milton is a Technical Specialist with Belcan Engineering, where he applies data science and machine learning to aerospace prognostics and health management. His work spans the development of end-to-end analytical systems, from large-scale flight and fleet data preparation and feature engineering through statistical analysis, model development, optimization, validation, and production integration. Drawing on a background in applied physics and data science, he is particularly interested in combining physical understanding, statistical reasoning, and machine learning to characterize complex system behavior and detect emerging anomalies. He is a named inventor on a patent application for a hybrid physics-informed analytics approach that uses electrostatic exit-debris sensor data to detect hot-section distress in jet engines. Milton holds a B.S. in Applied Physics and an M.S. in Data Science from the University of Connecticut. His research interests include physics-informed machine learning, resource-aware and onboard AI, scalable model development and optimization, and the validation and lifecycle assurance of machine-learning systems for aerospace applications.

Ge Zhou is an accomplished engineering leader with more than 20 years of experience in systems and software development, fuel cell technology, product design, diagnostics, and performance analysis. He currently serves as a Senior Technical Lead and Project Manager in Systems and Software Engineering at Belcan, a Cognizant company, where he provides technical leadership and drives the successful execution of engineering projects. Dr. Zhou earned his Bachelor of Science degree in Chemical Engineering from Tianjin University in Tianjin, China, in 1996, and his Ph.D. in Mechanical Engineering from the University of Iowa in Iowa City, Iowa, in 2007.