Unsupervised Fault Detection and Classification for Environmental Control and Life Support Systems Using Variational Autoencoders and Support Vector Machines

Ahsanul Abedin
MS in Industrial Engineering, Texas State University
San Marcos, TX, USA. Email: wjw61@txstate.edu

***Abstract*—Environmental Control and Life Support Systems (ECLSS) are mission-critical subsystems for crewed spaceflight. Failures in oxygen generation, carbon dioxide removal, or pressure regulation can rapidly compromise crew safety, especially during long-duration missions. This paper develops a hybrid machine-learning framework that combines a Variational Autoencoder (VAE) for unsupervised anomaly detection with a Support Vector Machine (SVM) for supervised fault-type classification, using a high-fidelity synthetic ECLSS dataset. The VAE is trained only on nominal multivariate time-series data and detects anomalies via reconstruction error thresholds, while the SVM operates on the learned latent space to classify five system-level fault modes. Experiments demonstrate effective anomaly separation (ROC–AUC ≈ 0.85) and high fault-type test accuracy ($\approx 95.6\%$) across five fault classes, indicating that VAE latent representations are highly informative for downstream classification. The results suggest that combining unsupervised representation learning with classical classifiers is a promising approach for ECLSS health monitoring.**

***Index Terms*—Environmental Control and Life Support Systems (ECLSS), anomaly detection, variational autoencoder (VAE), support vector machine (SVM), fault classification, synthetic dataset.**

I. Introduction

Deep-space and long-duration crewed missions depend critically on Environmental Control and Life Support Systems (ECLSS) to maintain breathable atmosphere, safe pressure, and acceptable carbon dioxide levels. ECLSS subsystems, such as the Oxygen Generation Assembly (OGA), Carbon Dioxide Removal Assembly (CDRA), and pressure control, exhibit nonlinear behavior, sensor noise, and potential component degradation. Undetected faults in these systems pose immediate risks to crew safety and mission success.

Traditional health monitoring in aerospace has relied heavily on physics-based models and threshold alarms. While effective for well-understood modes, these approaches may struggle under complex interactions, unmodeled dynamics, or subtle degradation trends. Data-driven methods, particularly deep learning, offer an alternative by learning patterns directly from sensor data. This project investigates a hybrid approach:

- A Variational Autoencoder (VAE) is trained only on nominal ECLSS cycles and used to detect anomalies via reconstruction error.
- A Support Vector Machine (SVM) is trained on VAE latent features to classify fault types for anomalous samples.

The primary objective is to design and evaluate a machine-learning pipeline that:

1) Learns normal ECLSS sensor behavior in an unsupervised fashion.
2) Detects deviations (faults) using VAE reconstruction error.
3) Classifies detected faults into interpretable fault types using an SVM.

The project addresses the following questions:

- Can a VAE trained exclusively on nominal multivariate sensor data reliably detect ECLSS anomalies using reconstruction error thresholds?
- Do the VAE latent representations provide sufficient separation to enable accurate fault-type classification using an RBF-kernel SVM?

The central hypothesis is that a β-VAE trained on nominal cycles will learn a compact, low-dimensional representation of normal ECLSS operation. Deviations from this manifold will induce elevated reconstruction errors, enabling anomaly detection. Furthermore, the latent mean vectors will encode fault-specific structure that an SVM can exploit to achieve high fault-type classification accuracy.

II. Literature Review

Recent work on deep space habitats (DSHs) emphasizes that long communication delays (minutes to over twenty minutes) make real-time ground intervention infeasible, forcing safety-critical subsystems such as ECLSS to operate with much greater Earth-independence [1]. Traditional threshold-based monitoring and ground-centric supervision are poorly suited to this regime, as seemingly minor faults (e.g., sensor drift, valve stiction, leaks) can rapidly escalate without timely detection [2]. NASA requirements therefore explicitly call for onboard,

autonomous anomaly detection and resolution, with human experts "on the loop" rather than "in the loop" [1].

Prognostics and Health Management (PHM) frameworks provide a natural foundation for such autonomy, integrating diagnostics and prognostics to track system health and anticipate degradation [3], [4]. In aerospace, PHM deployment has yielded tangible gains, for example, increasing UH-60 fully mission capable status from 65% to 87% while reducing unscheduled maintenance, and benefits from standardized integration architectures such as the Open System Architecture for Condition-Based Maintenance [3], [4]. Building on this, Ibrahim *et al.* proposed a generative ML framework tailored to cyclical ECLSS processes on DSHs, addressing model rigidity, manual retraining workflows, and scarce labeled fault data [1]. Their approach couples a VAE trained on nominal data for anomaly detection (via reconstruction error) with an SVM on latent features for fault diagnosis, achieving false-negative and false-positive rates below 10% and correct classification rates above 90% on the Simulation Testbed for Exploration Vehicle ECLSS dataset [1].

Parallel efforts investigate digital twins as a complementary route to ECLSS autonomy. Glaessgen and Stargel introduced the digital twin concept for NASA vehicles, fusing high-fidelity simulations, fleet history, and live sensor data into a continuously updated virtual replica [5]. For deep space ECLSS, such twins support uncertainty-aware prediction and autonomous decision-making during communication blackouts [2]. More broadly, ML research in aerospace fault detection demonstrates the effectiveness of deep sequence and generative models: LSTM-based detectors have been deployed on spacecraft telemetry streams [6], while deep variational autoencoders achieve dimensionality reduction with reconstruction-based detection rates above 99% for process faults [7]. Against this backdrop, the present project evaluates whether a VAE–SVM framework trained on synthetic ECLSS sensor data can reproduce prior performance targets (FNR/FPR $< 10\%$, CCR $> 90\%$) and thereby support autonomous fault management for DSHs operating under strict communication constraints [1].

III. Synthetic ECLSS Data Generation

To train and evaluate the anomaly–detection pipeline without access to proprietary flight data, we generated a fully synthetic Environmental Control and Life Support System (ECLSS) dataset. The generator follows the structure of Ibrahim *et al.* [1] and is parameterized to resemble International Space Station (ISS) cabin conditions for oxygen, carbon dioxide, and pressure ranges [8].

All codes are implemented in Python (Jupyter Notebook) and released in the project repository.[1] The final configuration used in this study is summarized in `DatasetConfig`. Unless otherwise noted, we use

- $n_{\text{classes}} = 6$ system modes,
- $n_{\text{samples/class}} = 120$ cycles per system mode (total $N = 720$ cycles),
- $T = 1000$ time steps per cycle,
- sampling rate $f_s = 2$ Hz (cycle duration ≈ 8.3 minutes),
- three sensors: O_2 concentration (%), CO_2 concentration (%), and cabin pressure (psi).

1) Nominal Adsorption–Desorption Cycle: For each cycle, we first generate a nominal trajectory $x_t \in \mathbb{R}^3$ for $t = 1, \ldots, T$, corresponding to the state vector

$$x_t = \begin{bmatrix} \mathrm{O_2}(t) \\ \mathrm{CO_2}(t) \\ P(t) \end{bmatrix}$$

Nominal conditions are centered around ISS-like setpoints [8] : $\mu_{\mathrm{O_2}} = 20.9\%$, $\mu_{\mathrm{CO_2}} = 0.3\%$ (corresponding to $\approx$ 2 mmHg), and $\mu_P = 14.7$ psi. Each sensor follows a low–amplitude periodic waveform representing the regular adsorption–desorption cycle:

$$\begin{aligned} \mathrm{O_2}(t) &= \mu_{\mathrm{O_2}} + A_{\mathrm{O_2}} \sin(2\pi f t), \\ \mathrm{CO_2}(t) &= \mu_{\mathrm{CO_2}} - A_{\mathrm{CO_2}} \sin(2\pi f t), \\ P(t) &= \mu_P + A_P \cos(2\pi f t), \end{aligned}$$

where $A_{\mathrm{O_2}} = 0.3$, $A_{\mathrm{CO_2}} = 0.10$, $A_P = 0.3$, and the frequency is jittered as $f = f_0 \cdot u$ with $u \sim \mathcal{U}(0.95, 1.05)$.

To emulate slow process drift and sensor aging, we add a small random walk to each channel:

$$\mathrm{O_2}(t) \leftarrow \mathrm{O_2}(t) + \sum_{\tau=1}^{t} \epsilon_{\mathrm{O_2},\tau}, \qquad \epsilon_{\mathrm{O_2},\tau} \sim \mathcal{N}(0, \sigma^2_{\text{drift,O}_2}),$$

$$\mathrm{CO_2}(t) \leftarrow \mathrm{CO_2}(t) + \sum_{\tau=1}^{t} \epsilon_{\mathrm{CO_2},\tau}, \quad \epsilon_{\mathrm{CO_2},\tau} \sim \mathcal{N}(0, \sigma^2_{\text{drift,CO}_2}),$$

$$P(t) \leftarrow P(t) + \sum_{\tau=1}^{t} \epsilon_{P,\tau}, \qquad \epsilon_{P,\tau} \sim \mathcal{N}(0, \sigma^2_{\text{drift,P}}),$$

followed by additive white sensor noise with standard deviations $\sigma_{\mathrm{O_2}} = 0.05$, $\sigma_{\mathrm{CO_2}} = 0.02$, and $\sigma_P = 0.05$. Finally, all trajectories are clipped to physically meaningful ranges: $\mathrm{O_2} \in [0, 100]\%$, $\mathrm{CO_2} \in [0, 5]\%$, and $P \in [0, 20]$ psi.

2) System-Level Fault Modes: On top of the nominal cycle, we inject one of six system modes, indexed by $c \in \{0, \ldots, 5\}$:

1) $c = 0$: Nominal (no system-level fault),
2) $c = 1$: CO_2 leak,
3) $c = 2$: valve stiction,
4) $c = 3$: vacuum anomaly,
5) $c = 4$: CDRA degradation,
6) $c = 5$: OGA degradation.

For a given mode c and severity parameter $s \in [0, 1]$, we construct a faulty cycle $\tilde{x}_t = f_c(x_t, s)$ as follows:

a) CO_2 leak ($c = 1$): We model a gradual CO_2 accumulation beginning at mid–cycle. For $t > T/2$, the CO_2 baseline is increased by a ramp of magnitude $\Delta_{\mathrm{CO_2}} \in [0.05, 0.40]\%$, interpolated from s:

$$\tilde{\mathrm{CO_2}}(t) = \mathrm{CO_2}(t) + r(t),$$

[1] See `src/generate_eclss_dataset.py`.

$$\text{r(t)} = \begin{cases} 0, & t \le T/2, \\ \text{linspace}(0,\, \Delta_{\text{CO}_2},\, T - T/2), & t > T/2. \end{cases}$$

b) Valve stiction ($c = 2$): Valve stiction is represented as sluggish pressure dynamics obtained by passing $P(t)$ through a first-order low–pass filter,

$$\tilde{P}(t) = \alpha\, \tilde{P}(t-1) + (1-\alpha)P(t),$$

where the smoothing factor $\alpha \in [0.85, 0.99]$ increases with severity, leading to visibly flattened pressure excursions.

c) Vacuum anomaly ($c = 3$): A vacuum anomaly corresponds to a localized Gaussian pressure drop near mid–cycle:

$$\tilde{P}(t) = P(t) - \Delta P \exp\left(-\frac{(t - T/2)^2}{2\sigma^2}\right),$$

with $\Delta P \in [0.5, 2.0]$ psi and width parameter σ chosen so that the anomaly occupies roughly 80 samples at low severity and narrows as s increases.

d) CDRA degradation ($c = 4$): CDRA degradation is modeled as a cycle–wide increase of the CO_2 baseline:

$$\tilde{\text{CO}}_2(t) = \text{CO}_2(t) + \Delta_{\text{CDRA}}, \qquad \Delta_{\text{CDRA}} \in [0.05, 0.5]\%.$$

e) OGA degradation ($c = 5$): Conversely, OGA degradation reduces the O_2 baseline:

$$\tilde{\text{O}}_2(t) = \text{O}_2(t) - \Delta_{\text{OGA}}, \qquad \Delta_{\text{OGA}} \in [0.5, 4.0]\%.$$

In all modes, the severity parameter s is scheduled to increase linearly with the sample index k within each class and perturbed with Gaussian noise, as implemented in `generate_dataset`. This produces a continuum of mild to severe faults rather than discrete levels.

3) Sensor-Level Fault Injection: In addition to system faults, we randomly inject sensor pathologies into 15% of cycles (`p_sensor_fault = 0.15`). The following sensor fault types are implemented:

1) No sensor fault,
2) Bias drift: a linear drift on a single sensor channel over the cycle,
3) High noise: increased Gaussian noise ($\times 4$) on one or more channels,
4) Partial freeze: a contiguous segment where one sensor is held constant,
5) Spike outliers: random positive or negative spikes on a single channel.

The function `apply_random_sensor_fault` chooses one of these (excluding "None") and applies it after system-level faults. All signals are then re–clipped to the physical ranges in Table I.

4) Safety Flags and Metadata: For each final cycle, we compute simple NASA-inspired safety flags that indicate whether ISS-like warning or critical limits are violated [8]. Thresholds include:

- CO_2 warning and critical maxima (0.7% and 1.0%),
- low O_2 warnings (mean $O_2 < 19\%$) and critical ($< 16\%$),
- pressure warning band ($14.0 \le \bar{P} \le 15.4$ psi) and critical band ($13.5 \le \bar{P} \le 15.8$ psi).

The resulting Boolean flags (e.g., `flag_CO2_warn`, `flag_P_crit`) are stored in a metadata table together with system class, sensor fault type, and severity parameter for each cycle.

TABLE I
SENSOR RANGES USED FOR CLIPPING SYNTHETIC TRAJECTORIES.

Sensor	Min	Max	Units
O_2	0	100	%
CO_2	0	5	%
Pressure	0	20	psi

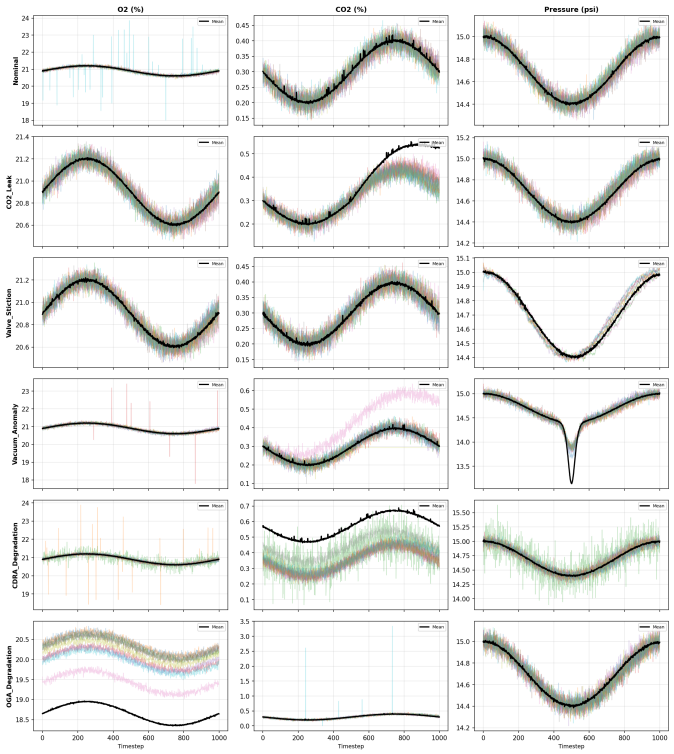

Fig. 1. Overlay of ECLSS cycles for each system class, showing O_2, CO_2, and pressure trajectories highlighting qualitative differences between nominal and faulty modes.

5) Dataset Validation and Visual Inspection: We perform a multi–step validation routine before using the data for model training:

1) Physical and numeric checks: verify that all samples satisfy the clipping bounds, and that no NaN or Inf values are present.
2) Temporal coherence: compute lag–1 autocorrelation for a subset of cycles and sensors; the mean autocorrelation is required to exceed 0.7, indicating smooth, physically plausible time series.
3) Class separability: flatten each cycle, project to two dimensions using PCA, and compute the silhouette score over system classes; scores above 0.2 suggest that the modes are at least moderately separable in feature space.

4) Overlay plots: for each system class, we overlay up to ten cycles per sensor and superimpose the class–wise mean trajectory to visually confirm that the injected faults match their intended signatures.

Additional summary statistics (class counts, fault distributions, sensor ranges, and configuration values) are automatically exported as a Markdown report (`dataset_summary.md`) and stored alongside the NumPy arrays (`cycles_3d.npy`, `labels_system.npy`, `labels_sensor.npy`) for downstream VAE and SVM training.

IV. Data Pre-Processing

The preprocessing pipeline is designed to avoid information leakage and to reflect realistic deployment:

A. Nominal/Fault Partitioning

All samples with system class 0 (nominal) are reserved for unsupervised VAE training and anomaly threshold calibration. Samples from classes 1–5 are treated as faulty and used only for SVM fault-type training and testing.

B. Train/Validation/Test Splits

Nominal samples are split as follows:

- 70% for VAE training
- 15% for validation (early stopping)
- 15% for nominal test data

Faulty samples (classes 1–5) are split with stratification:

- 70% for SVM training
- 30% for fault-type testing

A combined anomaly detection test set is built by concatenating nominal test samples and faulty test samples, and labeling them as 0 (nominal) or 1 (anomaly).

C. Normalization and Flattening

A single *StandardScaler* is fitted using only the nominal training data. This scaler is then applied to all other splits. After scaling, each $(N, T, 3)$ time series is flattened into a 3000-dimensional vector, suitable for dense neural networks.

V. Methods

A. Variational Autoencoder Architecture

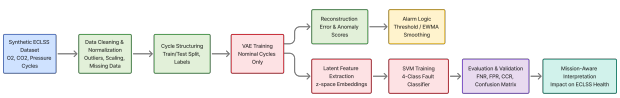

Fig. 2. Overview of the proposed pipeline: (1) synthetic ECLSS sensor data generation, (2) VAE training on nominal cycles, (3) anomaly detection using reconstruction error, and (4) fault-type classification in latent space using an RBF-kernel SVM.

The Dense VAE operates on 3000-dimensional standardized vectors. The encoder consists of:

- Fully connected layers: $3000 \to 1024 \to 512 \to 256$
- Each layer uses Batch Normalization, LeakyReLU activation, and Dropout.

From the final hidden layer, two linear heads compute the latent mean μ and log-variance $\log \sigma^2$ for a 32-dimensional latent vector z. The decoder mirrors the encoder structure, mapping z back to a reconstructed 3000-dimensional vector $\hat{x}$.

The loss function combines reconstruction error and KL divergence:

$$\mathcal{L} = \text{MSE}(x, \hat{x}) + \beta \cdot D_{\text{KL}}(q_\phi(z|x) \,\|\, p(z)), \qquad (1)$$

with $\beta = 0.3$ and $p(z)$ taken as a standard normal prior.

Training uses the Adam optimizer with learning rate 10^{-3}, batch size 8, and early stopping based on validation loss.

B. Anomaly Detection via Reconstruction Error

For each sample, the VAE computes reconstruction error as mean squared error across all features. A scalar threshold τ is set as the 99th percentile of reconstruction errors on nominal training data. At deployment, any sample with error above τ is flagged as anomalous.

C. Latent-Space Fault Classification with SVM

Latent mean vectors μ from the encoder are used as low-dimensional features for fault-type classification. Only faulty samples (classes 1–5) are considered for this task. Features are standardized, and an RBF-kernel SVM is trained with hyperparameters selected via 5-fold cross-validation over a small grid of C and γ values.

VI. Results and Conclusion

A. Anomaly Detection Performance

On the combined anomaly detection test set (nominal + faulty):

- The ROC–AUC of reconstruction error vs. binary label is approximately 0.849.
- Using the 99th-percentile threshold from nominal training data yields a high nominal detection rate (train nominal accuracy $\approx 98.8\%$) with a modest anomaly recall on the test set.

B. Latent Space Visualization

Latent representations are projected into two dimensions using t-SNE or UMAP to visually inspect class separation. Nominal and faulty samples form distinct clusters, and different fault types occupy partially separated regions in latent space.

C. Fault-Type Classification Performance

Using the VAE latent means as features, the RBF-kernel SVM achieves:

- Training accuracy: $\approx 99.5\%$
- Cross-validation accuracy: $\approx 95.5\%$
- Test accuracy on faulty samples: $\approx 95.6\%$

Per-class precision, recall, and F1-scores are all high, with only minor confusion between certain fault pairs.

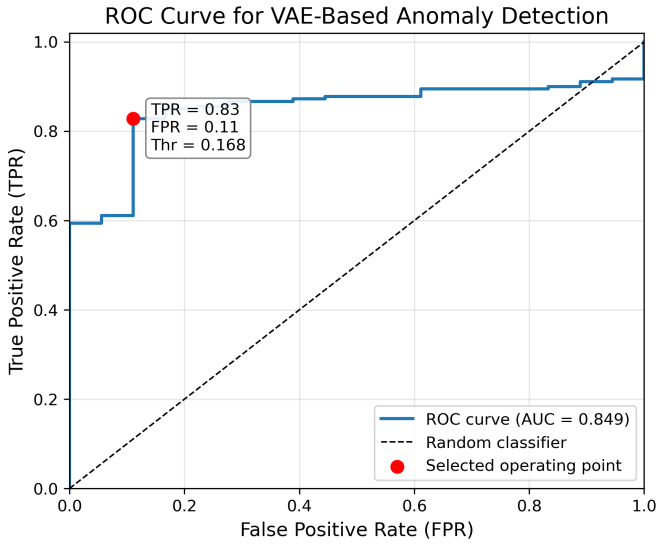

Fig. 3. Receiver Operating Characteristic (ROC) curve for anomaly detection using VAE reconstruction error, including the area under the curve and a highlighted operating point corresponding to the chosen threshold.

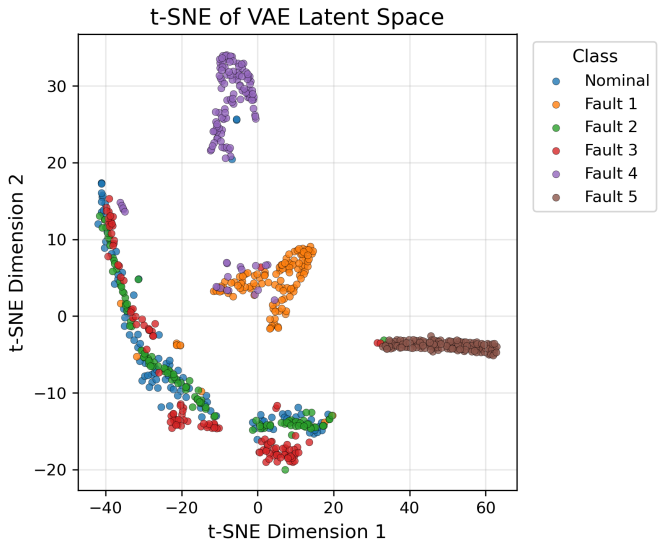

Fig. 4. t-SNE visualization of VAE latent mean vectors, colored by system class. The plot illustrates separation between nominal operation and the five simulated fault modes.

Table II summarizes the per-class precision, recall, and F1-scores achieved by the SVM classifier on the held-out faulty test set. All fault categories exhibit high classification accuracy, with F1-scores ranging from 0.928 to 1.000. Recall values exceed 0.89 for all classes, indicating effective sensitivity across fault modes, while precision exceeding 0.92 demonstrates a low false classification rate. The consistent performance across uniformly sized classes supports classifier robustness and the absence of class imbalance bias.

TABLE II
SVM FAULT-TYPE CLASSIFICATION PERFORMANCE

Fault Class	Precision	Recall	F1-score	Support
1	0.9231	1.0000	0.9600	36
2	0.9189	0.9444	0.9315	36
3	0.9697	0.8889	0.9275	36
4	0.9714	0.9444	0.9577	36
5	1.0000	1.0000	1.0000	36

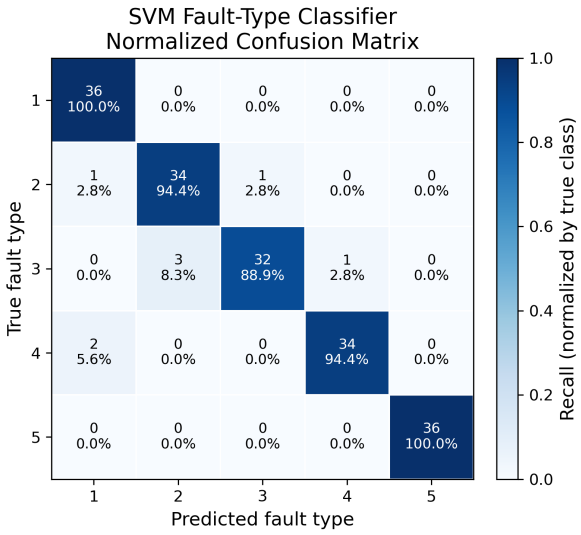

Fig. 5. Normalized confusion matrix for the SVM fault-type classifier in VAE latent space. Rows correspond to true fault classes and columns to predicted classes.

D. Learning Curve and Bias–Variance Analysis

Learning curves and cross-validation metrics indicate a small but acceptable generalization gap between training and validation performance, suggesting mild overfitting but overall good generalization.

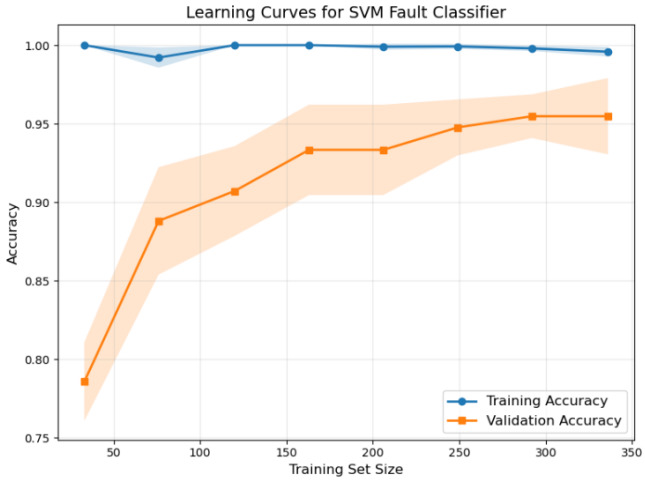

Fig. 6. Learning curve for the SVM fault classifier, showing training and cross-validation accuracy as a function of the number of training samples.

E. Discussion and Conclusion

The results support the original hypothesis:

- A VAE trained solely on nominal data can successfully distinguish anomalous ECLSS behavior based on reconstruction error.
- The latent representations learned by the VAE are discriminative for fault diagnosis, enabling an SVM classifier to achieve approximately 95.6% test accuracy across five simulated fault modes.

These findings confirm that unsupervised representation learning followed by supervised fault classification is a viable strategy for ECLSS health monitoring, particularly in scenarios where labeled fault data are limited but nominal operational data are abundant. Following anomaly detection, the SVM achieved strong and consistent fault-type discrimination, with precision exceeding 92% and recall above 88% for all classes. F1-scores ranged from 0.93 to 1.00, yielding a macro-averaged F1 of approximately 95.5%. Balanced supports of 36 test samples per class, ensured by stratified splitting of the synthetic dataset, enabled unbiased evaluation and demonstrated robustness across fault types.

Together, the VAE and SVM form a complementary detection–diagnosis architecture well suited for deep-space autonomy, where unsupervised detection identifies abnormal system behavior and supervised classification provides actionable fault attribution for maintenance planning and logistics optimization.

F. Future Work

Although the results are strong, the dataset remains synthetically generated and therefore lacks certain complexities inherent to real spacecraft telemetry, including sensor drift accumulation, correlated measurement faults, mixed fault conditions, and gradual degradation phenomena. Future work will extend validation to higher-fidelity or real mission data, develop sequence-aware models, and integrate diagnostic outputs into closed-loop reliability and resource management frameworks. Future extensions include:

- Incorporating temporal models (e.g., sequence VAEs or Transformers) instead of flattening time series.
- Exploring adaptive or Bayesian thresholding schemes for anomaly detection.
- Testing the framework on real ECLSS or ISS telemetry when available.
- Integrating anomaly detection outputs with decision-making models for maintenance scheduling and mission planning.

Acknowledgment

The author thanks the faculty and colleagues at the Ingram School of Engineering, Texas State University, for their guidance and feedback during this project.

References

[1] A. Ibrahim *et al.*, "Autonomous anomaly response for environmental control and life support systems using generative machine learning," *Computers & Chemical Engineering*, vol. 190, 2026.

[2] A. Gratius *et al.*, "Digital twins for deep-space habitat autonomy and health management," *Acta Astronautica*, vol. 219, pp. 48–62, 2024.

[3] L. Li, W. Zhou, and Y. Song, "Prognostics and health management for aerospace systems: State of the art," *IEEE Trans. Reliab.*, vol. 66, no. 2, pp. 414–425, 2017.

[4] K. Nguyen, J. Medjaher, and M. Zolghadri, "PHM-based maintenance optimization for aerospace systems," *Reliability Engineering & System Safety*, vol. 184, pp. 24–35, 2019.

[5] E. H. Glaessgen and D. S. Stargel, "The digital twin paradigm for future NASA and U.S. Air Force vehicles," in *Proc. AIAA Modeling and Simulation Technologies Conf.*, 2012.

[6] K. Hundman *et al.*, "Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding," in *Proc. ACM KDD Int. Conf. on Knowledge Discovery & Data Mining*, 2018.

[7] A. San Martin, J. V. Draper, and H. M. Pandit, "Deep variational autoencoders for multivariate fault detection in industrial processes," *Computers & Chemical Engineering*, vol. 126, pp. 309–318, 2019.

[8] L. Chen, C. Reuland, J. Rivas, W. Misek, and A. Rangel, "Unveiling the system-of-systems complexity in regenerative ECLSS: insights from ISS," in *Proc. 54th Int. Conf. on Environmental Systems (ICES)*, Prague, Czech Republic, Jul. 2025.