Anomaly Detection In Space Logs

Dr. Lincy Meera Mathews
Dept. of Information Science and Engineering
Ramaiah Institute of Technology
Bengaluru, India
lincymm99@gmail.com

Amith M
Dept. of Information Science and Engineering
Ramaiah Institute of Technology
Bengaluru, India
1ms21is017@msrit.edu

Anirudh S
Dept. of Information Science and Engineering
Ramaiah Institute of Technology
Bengaluru, India
1ms21is018@msrit.edu

Akhilesh
Dept. of Information Science and Engineering
Ramaiah Institute of Technology
Bengaluru, India
1ms21is014@msrit.edu

Agniv Chakraborty
Dept. of Information Science and Engineering
Ramaiah Institute of Technology
Bengaluru, India
1ms21is013@msrit.edu

***Abstract*—Effective anomaly detection in spacecraft telemetry is critical for mission safety amid increasing system complexity and data volume. Traditional methods struggle with high-dimensional, unstructured telemetry logs. This paper presents a modular and scalable anomaly detection system based on the Deep Bi-Level Contrastive Learning (DBLCL) framework, tailored for spacecraft telemetry. The approach integrates data preprocessing, representation learning via contrastive self-supervised learning, anomaly scoring with Gaussian Mixture Models and Isolation Forests, and dynamic thresholding for adaptive detection. Unstructured logs are embedded using transformer models, enabling robust feature extraction without labeled anomalies. Evaluated on the NASA Spacecraft Telemetry Anomaly Dataset, DBLCL demonstrates strong detection performance and flexibility for diverse aerospace applications.**

***Index Terms*—Anomaly Detection, Spacecraft Telemetry, Contrastive Learning, Deep Learning, Gaussian Mixture Model, Isolation Forest, Unstructured Logs, Ensemble Methods**

I. Introduction

The increasing complexity of spacecraft systems has led to a rise in in-orbit failures, threatening mission safety and reliability. For instance, satellite failure rates in China increased from 0.29 before the 1990s to 1.12 in the early 2000s. Anomaly detection, the process of identifying unusual or unexpected events in spacecraft telemetry logs, is critical for early fault diagnosis and prevention of mission failures .

Space missions generate massive volumes of telemetry logs capturing real-time health and behavior of onboard systems. These logs, often multivariate time series from diverse subsystems, are the primary source for ground-based anomaly detection. However, their high dimensionality, unstructured nature, and non-stationary patterns—exacerbated by unpredictable deep space environments—pose significant challenges for effective anomaly detection .

Anomalies in telemetry data may indicate sensor degradation, hardware malfunctions, or environmental disturbances. Early detection is vital to avoid catastrophic failures and ensure mission continuity. Traditional manual or threshold-based methods struggle with the scale and complexity of modern telemetry, especially for spacecraft like Tianwen-1, which generates over 10,000 telemetry parameters across subsystems .

A. Motivation and Scope

Spacecraft generate vast amounts of telemetry data across numerous subsystems, making manual anomaly detection impractical. Early detection of anomalies is crucial, as unusual patterns often signal impending system failures. Traditional rule-based and supervised methods struggle with the scarcity of labeled anomalies and the unstructured nature of many logs, while fixed-threshold approaches often produce false positives due to context-dependent variations and transient space events.

This work proposes a scalable anomaly detection system based on Deep Bi-Level Contrastive Learning (DBLCL), which leverages contrastive learning to extract robust features from both structured telemetry and unstructured logs using transformer-based embeddings. An ensemble of Gaussian Mixture Models and Isolation Forests generates anomaly scores, and a dynamic thresholding mechanism adapts detection sensitivity to varying data characteristics. This approach addresses the challenges of high-dimensional, heterogeneous data, improving detection accuracy and robustness across diverse spacecraft telemetry scenarios.

B. Issues and Challenges

Following issues and challenges were faced while working on this project:

- **Data Volume and Dimensionality:** Modern spacecraft generate thousands of telemetry parameters, producing massive volumes of data beyond human monitoring capacity. High-dimensional, multivariate data require complex analysis and correlation across many channels, increasing computational demands. Integration of unstructured and heterogeneous data, such as engineer reports and alert logs, alongside structured sensor readings adds further complexity.

- **Rarity and Imbalance of Anomalies:** Spacecraft are designed for robustness, resulting in sparse anomalies; for example, fewer than 200 critical anomalies were reported across seven spacecraft over a decade. The rarity of anomalies causes severe class imbalance, complicating supervised model training and increasing false positive and false negative rates.
- **Data Availability and Privacy:** There is a lack of publicly available, labeled spacecraft anomaly datasets, limiting reproducibility and benchmarking. Proprietary and sensitive operational data restrictions further reduce access to sufficient training and validation data.
- **Dynamic and Non-Stationary Environments:** Spacecraft telemetry is inherently non-stationary, with data patterns changing due to operational modes, system aging, and environmental factors. Deep space conditions are unpredictable and volatile, with radiation, thermal cycles, and other factors causing unexpected system behaviors.

C. Problem Statement

The rapid advancement of spacecraft systems has led to an explosion in the volume and complexity of telemetry and log data required for monitoring and anomaly detection. Traditional methods, which often rely on threshold-based or manual analysis, are increasingly inadequate due to the unstructured nature of logs, high dimensionality of telemetry, and the rarity of anomalies, which result in imbalanced datasets [1] [2] [3].

Current anomaly detection techniques [4] [2] [3].struggle with these challenges, often producing high false positive or negative rates, lacking the ability to integrate unstructured data, and failing to adapt to new patterns without extensive retraining.

D. Proposed Model

We propose the Deep Bi-Level Contrastive Learning (DBLCL) framework, a scalable anomaly detection system tailored for spacecraft telemetry and log data. The framework preprocesses both structured telemetry and unstructured logs, embedding the latter using a transformer-based text encoder to produce uniform dense vector representations. Data augmentation techniques, such as Gaussian noise addition, create augmented pairs used in contrastive learning with the NT-Xent loss, enabling the model to learn robust and discriminative features invariant to minor perturbations. A bidirectional LSTM encoder captures temporal dependencies and contextual information by processing sequences in both directions, outputting fixed-length embeddings that effectively represent normal operational patterns.

Anomaly detection is performed by combining Gaussian Mixture Models (GMM) for density estimation and Isolation Forests for isolation-based scoring on the learned embeddings. The ensemble approach leverages complementary strengths to improve detection accuracy. A dynamic thresholding mechanism optimizes the anomaly decision boundary based on validation data, adapting to changing data distributions and operational modes to maintain high performance. This end-to-end pipeline efficiently processes large volumes of telemetry data, offering a robust and flexible solution for real-time anomaly detection in mission-critical spacecraft environments.

II. LITERATURE REVIEW

Based on recent advancements in spacecraft telemetry anomaly detection, several innovative methodologies have emerged, enhancing the capabilities of deep learning, density estimation, and self-supervised learning techniques. Below is an elaborated literature survey incorporating these developments:

A 2024 study [5] conducted an extensive evaluation of various deep learning architectures, including CNNs, RNNs, LSTMs, and Transformers, specifically applied to spacecraft telemetry datasets. Their methodology began with statistical profiling for telemetry segmentation, followed by sequence modeling techniques to capture temporal dependencies. The study revealed valuable insights into each model's strengths and weaknesses in handling mission variability, sparse data, and onboard constraints. CNNs performed well on structured signals, while LSTMs better captured long-range temporal patterns. However, Transformer models showed promise for parallel processing despite higher resource demands. The authors emphasized the importance of adaptability and inductive biases, particularly in low-label regimes. Despite their merits, these models struggled with generalization, requiring significant tuning.

[6] proposed DAEDL, a semi-supervised framework that enhances Evidential Deep Learning (EDL) through Gaussian Discriminant Analysis (GDA), enabling better modeling of predictive uncertainty. The model employs Dirichlet distributions to estimate both in-distribution and out-of-distribution samples, improving the robustness of anomaly classification. By embedding density-awareness, DAEDL distinguishes subtle anomalies in high-dimensional spaces. The integration of GDA into EDL allows for the refinement of latent feature distributions, improving decision boundaries. This method proved effective in scenarios with limited labeled data and high-class imbalance. However, DAEDL's dependency on accurate density estimation introduces computational challenges. Its efficacy is also limited to classification tasks, reducing flexibility for regression-based applications.

[7] introduced a model combining Temporal Convolutional Networks (TCNs) with Dynamic Graph Attention for spacecraft anomaly detection. This hybrid architecture captures inter-sensor dependencies while preserving temporal granularity, enabling more accurate detection of contextual anomalies. The use of graph attention mechanisms enhances the model's sensitivity to variable interactions. TCNs offer long receptive fields and avoid gradient vanishing issues typical in RNNs, contributing to stable training. The study demonstrated strong performance across multivariate telemetry datasets, outperforming traditional RNN and LSTM approaches. However, the complexity of this architecture poses deployment challenges, particularly for resource-constrained environments. The extensive training time

required by long temporal dependencies also limits its real-time applicability thus affecting our usecase.

[8] explored a classical density-based outlier detection approach applied to NASA's Kepler space telescope data. They utilized k-Nearest Neighbor (k-NN) algorithms in combination with Principal Component Analysis (PCA) to reduce feature dimensionality and identify anomalies. The study targeted astrophysical anomalies by detecting low-density regions in the transformed feature space. The simplicity of k-NN allowed for interpretability and ease of implementation. However, the method exhibited limitations in high-dimensional telemetry data due to distance metric distortion, a phenomenon known as the curse of dimensionality. Additionally, the approach had a high computational cost, limiting its use in real-time or onboard settings. The paper emphasized the trade-off between model interpretability and performance in high-complexity data scenarios and hence having a significantly higher training time.

[9] developed an explainable anomaly detection method using a hybrid of LSTM networks and classical signal processing techniques like Short-Time Fourier Transform (STFT) and moving averages. This approach prioritized interpretability by providing visual insights into anomaly patterns, facilitating human-in-the-loop validation. The LSTM component enabled modeling of temporal dependencies, while STFT captured frequency-based characteristics of telemetry signals. The hybrid model was evaluated on multiple spacecraft telemetry datasets and demonstrated strong performance in identifying known anomalies. However, the approach assumed some prior knowledge of data distribution, which limits its effectiveness in fully unsupervised settings. Furthermore, it showed reduced sensitivity to subtle or contextual anomalies.

[10] introduced DCLOP, a deep clustering-based framework for detecting anomalies in satellite health monitoring systems. The model combined deep embedding learning with probabilistic outlier scoring, aiming to detect early signs of component degradation and operational anomalies. By learning compact latent representations of telemetry data, DCLOP facilitated improved clustering of normal versus abnormal behaviors. The probabilistic scoring mechanism enabled dynamic thresholding for anomaly alerts. While the approach showed high sensitivity in identifying deviations, it required clean and noise-free input data for effective embedding learning. The model's latent nature also posed challenges in terms of interpretability, especially for mission operators. Additionally, the computational demands of deep clustering made it less feasible for onboard applications.

[11] presented a fast anomaly detection method using an Extreme Learning Machine (ELM) optimized via the Grey Wolf Optimizer (GWO). The ELM model featured randomly initialized hidden neurons, enabling rapid training, while GWO was employed for hyperparameter selection to enhance detection performance. The approach offered low-latency inference, making it suitable for near-real-time applications. Experimental evaluations on spacecraft telemetry datasets showed promising results in identifying known anomalies. However, the fixed-interval confidence mechanism reduced its adaptability to changing operational conditions. ELM's random initialization also led to unstable generalization on non-stationary and multivariate telemetry data.

[11] proposed a Sparse Feature-Based Anomaly Detection (SFAD) model using K-SVD for sparse representation and One-Class SVM for anomaly scoring. The approach aimed to capture hybrid anomalies by constructing dictionary-based representations of telemetry data. Sparse coding emphasized key signal components, while the One-Class SVM model distinguished normal patterns from outliers. The method showed strong results on benchmark spacecraft telemetry datasets, particularly in identifying pattern deviations. However, its performance was highly sensitive to dictionary construction and tuning parameters. Improper selection led to increased false positives. Moreover, the local nature of sparse representation limited its ability to detect anomalies evolving over longer timeframes.

[12] applied conformal prediction techniques to anomaly detection by associating confidence measures with anomaly scores derived from density-based methods. The model processed univariate telemetry features and used conformal scoring to calibrate anomaly thresholds, providing interpretable and reliable alerts. This method improved transparency by attaching probabilistic guarantees to predictions, increasing user trust in automated decisions. However, the approach's reliance on handcrafted features limited its scalability to multivariate systems. Sensitivity to distance metrics also affected stability, particularly in noisy environments.

[13] presented LogBD, a BERT-based anomaly detection framework that incorporates Temporal Convolutional Networks (TCNs) for cross-domain log analysis. The method embeds log sequences into hyperspherical vector spaces to align distributions from different domains, facilitating transfer learning. This is particularly useful in spacecraft systems where logs vary across missions but share structural similarities. Computational costs were also significant due to the combined use of BERT and TCNs. Despite these drawbacks, the approach marked progress in extending anomaly detection capabilities beyond fixed-domain settings, promoting adaptability and reuse.

[14] introduced a self-supervised learning (SSL) model based on Directed Acyclic Graphs (DAGs) to detect anomalies in scientific workflows. The model leveraged graph neural networks and contrastive learning to encode workflow structure and behavior. Structural augmentations and masked modeling techniques enabled learning from unlabeled data, addressing the challenge of data scarcity. This approach achieved promising results in structured systems where workflow dependencies were clearly defined. However, the model's applicability to spacecraft telemetry, which is often asynchronous and less structured, remained limited. Additionally, the model required considerable computational resources, making it unsuitable for onboard deployment. The study highlighted the value of structural representations in anomaly detection but acknowledged the limitations of applying workflow-specific models to free-form telemetry data.

[15] proposed SSLPDL, a self-supervised anomaly detection method combining probabilistic labeling with masked signal

modeling for rare-event prediction. The model integrates generative and discriminative objectives, improving sensitivity to extreme events in highly imbalanced datasets. SSLPDL leverages temporal masking and pseudo-labeling to learn from unannotated data. The approach was validated on simulated and real-world telemetry scenarios, demonstrating improved detection of subtle anomalies. However, the model struggled with dynamic environments where anomaly characteristics evolved over time. Overfitting to skewed distributions was also observed, requiring regularization strategies. Additionally, the integration of spatiotemporal dependencies remained an open challenge. Despite these limitations, SSLPDL presented a promising direction for anomaly detection in complex, low-label regimes.

The next paper [16] introduces a recursive, kernel density estimator-based clustering algorithm designed to identify the entire cluster tree of a distribution by estimating its level sets without heavily relying on smoothness assumptions like Hölder continuity. Instead, the method depends on more intuitive geometric conditions, such as an inner cone condition and a thickness assumption, which ensure the robustness of level set estimation around critical levels where clusters form or split. The algorithm adaptively chooses the kernel bandwidth in a data-driven manner, allowing it to perform well even when the true density's smoothness is unknown. It can handle densities that are discontinuous, such as step functions, and provides finite sample guarantees, establishing consistency and convergence rates under these more flexible conditions. However, the approach assumes that densities are well-behaved in terms of their topology, like having finitely many split levels, and relies on certain regularity conditions that, while more general than smoothness, still restrict applicability to scenarios where the density's geometric features are sufficiently regular. Additionally, the analysis and guarantees focus primarily on estimating the cluster tree and split levels, rather than directly addressing the estimation of individual clusters in cases where the distribution contains a single, non-splitting cluster. The method also does not currently extend to estimating level sets in settings with highly irregular or fractal-like densities, and the choice of parameters like the initial bandwidth still influences performance, although the adaptive strategy mitigates this to some extent.

Another paper [17] introduces a Kernel Density Estimation (KDE) based sampling method to tackle imbalanced class distributions, a common issue in fields like fraud detection and medical diagnostics where the minority class is crucial. Traditional methods like undersampling and oversampling have drawbacks, such as information loss or overfitting. KDE offers a statistically sound approach by estimating the minority class's probability density and generating new instances. This involves using KDE with a Gaussian kernel and Scott's rule for bandwidth selection. New instances are created by sampling from the KDE, effectively 'spraying' around existing minority class points. Numerical experiments demonstrate that KDE can outperform methods like Random Oversampling (ROS), SMOTE, ADASYN, and NearMiss, often showing superior F1-scores and G-means with KNN and SVM classifiers. The method's flexibility lies in the choice of kernel functions and bandwidth, offering customization of the sampling process. However, limitations include sensitivity to bandwidth selection (potentially improved via cross-validation), computational complexity in high-dimensional spaces, potential performance dips due to the curse of dimensionality, and classifier-dependent performance, as seen with the MLP classifier. While tested on 12 real-life datasets, a broader evaluation could further validate its effectiveness. The paper suggests KDE is a valuable tool for addressing imbalanced class distribution problems, offering a balance between statistical rigor and practical performance.

III. Framework and System Design

The design of the anomaly detection system based on the Deep Bi-Level Contrastive Learning (DBLCL) framework is both modular, hierarchical and enabling to high scalability and adaptability to various spacecraft telemetry logging scenarios.

The overall system is broken down into four primary steps in the pipeline namely, Data Pipeline, Representation Learning, Anomaly Scoring Module, and Threshold-based Detection Engine—each with distinct responsibilities and tight integration to ensure end-to-end unsupervised anomaly detection.

A. System Architecture Overview

1) Data Pipeline and Preprocessing: The input to the system is a dataset containing logs from NASA's system records. These logs may be either structured (numerical features) or unstructured (textual log messages). The preprocessing pipeline adapts to both formats:

If the dataset includes a message column (unstructured logs), each log entry is transformed into a numerical embedding using a pre-trained transformer model (all-MiniLM-L6-v2 from SentenceTransformers). This model generates dense vector representations that capture the semantic meaning of the text.

If the data is already structured, the numeric columns are used directly after dropping any irrelevant or non-numeric features.

In both cases, the resulting feature vectors are standardized using StandardScaler to normalize the range and variance of each dimension. The normalized features are then reshaped to simulate a sequence of length one per sample, resulting in input tensors of shape [batchSize, sequenceLength=1, featureDim]. This formatting is necessary for compatibility with sequence-based learning components used later in the model.

This step of the pipeline ensures that all downstream modules of logs receive clean, normalized, and uniformly structured high-dimensional input vectors.

2) Feature Representation Learning using Contrastive Learning: At the core of the system lies a neural architecture designed to learn meaningful representations of logs. The architecture consists of two parts: an encoder and a projector.

Encoder Network

- The encoder is a fully connected feedforward network composed of two linear layers with ReLU activations and dropout regularization. This module transforms the

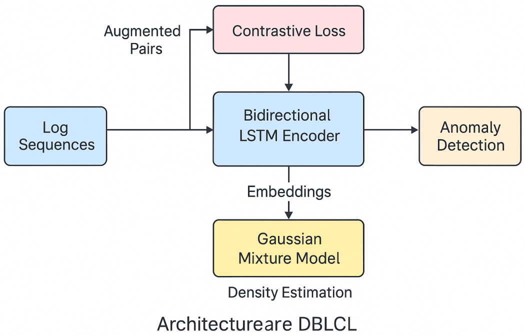

Fig. 1: DBLCL Architecture

input log vectors into a latent embedding space (h). The purpose of the encoder is to extract patterns that are consistent across similar log entries while being sensitive to significant structural deviations.

Projector Network Following the encoder is a projector module, also composed of two dense layers, that maps the latent representations (h) into a new space (z) used specifically for contrastive learning. This space is not used for anomaly scoring but helps the encoder learn robust representations.

Contrastive Learning Mechanism To train the encoder and projector without labels, we apply contrastive learning using the NT-Xent (Normalized Temperature-scaled Cross Entropy) loss. For each log entry in a training batch, two views are generated:

The original log vector.

A slightly perturbed version, created by adding small Gaussian noise (data augmentation).

These pairs are assumed to be "similar" (i.e., representing the same underlying class — typically a non-anomalous behavior). The contrastive loss function encourages the model to bring similar pairs (z1, z2) closer together in the projection space while pushing away representations of other entries in the batch.

This method is self-supervised, as it relies on synthetic pairing through augmentation and does not require human-labeled anomalies.

$$\mathcal{L}_{\text{InfoNCE}} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} 1_{[k \neq i]} \exp(\text{sim}(z_i, z_k)/\tau)} \quad (1)$$

- A large number of negative samples are constructed by selecting other samples in the mini-batch.
- This unsupervised loss function drives the network to create dense, well-separated clusters of normal patterns, improving downstream anomaly separability.

B. Embedding Extraction for Anomaly Detection

After contrastive training, the projector is no longer used. Instead, the encoder is used to generate embeddings (h) for each log entry. These embeddings are assumed to represent the typical behavior of the logs.

- To detect anomalies, we measure how well each new log fits into the distribution of these learned embeddings using two different models: Gaussian Mixture Model (GMM) and Isolation Forest.

1) Density-Based Anomaly Detection: As shown in Figure 1 anomaly detection is done using **Gaussian Mixture Model** and **Isolation Forest** .

Gaussian Mixture Model (GMM)

The GMM is a probabilistic model that assumes that the data is generated from a mixture of several Gaussian distributions. After fitting the GMM to the embeddings of the training data, it computes the log-likelihood of each sample — essentially, how likely it is to belong to the learned distribution.

Low log-likelihood values indicate that a log entry is not well represented by the model and is thus considered anomalous.

Isolation Forest

In parallel, an isolation forest is trained on the same embeddings. This model is an ensemble of decision trees that isolates anomalies based on how quickly they can be separated from the rest of the data. Anomalies tend to be more easily isolated, requiring fewer splits in the decision trees.

This model is particularly effective in high-dimensional spaces and complements the GMM by detecting samples that are structurally different from normal data.

2) Anomaly Score Fusion and Thresholding: To improve the robustness and accuracy of anomaly detection, the system combines scores from both GMM and Isolation Forest.

Both score distributions are normalized to a [0, 1] range to allow fair combination.

The final anomaly score is computed as a weighted sum of the two:

$$\text{Anomaly Score} = \alpha \cdot (1 - \text{GMM\_Score}) + (1 - \alpha) \cdot \text{IForest\_Score} \quad (2)$$

α is empirically tuned (By default, α = 0.6).

Here, GMM's log-likelihood is inverted since lower values indicate greater anomaly.

This fusion approach leverages the probabilistic confidence of GMM and the structural separation power of Isolation Forest.

To convert continuous anomaly scores into binary decisions (anomaly or not anomaly), the system uses percentile-based thresholding:

- A threshold is chosen such that any sample with a score above it is classified as an anomaly (-1), and the rest as not anomaly (1).
- The optimal percentile is determined through evaluation on a validation set, maximizing the F1-score.

3) Evaluation and Prediction: With the threshold finalized, the model is applied to the test set. For each test sample:

- The encoder generates its embedding.
- The GMM and Isolation Forest compute anomaly scores.
- The final score is calculated using the same weighted formula.
- A binary label is assigned using the selected threshold.

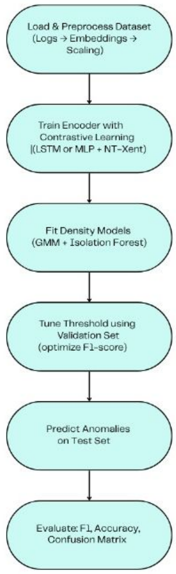

Fig. 2: Anomaly Detection Architecture flowchart

IV. Implementation

The proposed approach for anomaly detection in spacecraft telemetry data is based on a Deep Bi-Level Contrastive Learning (DBLCL) framework. This framework was designed to overcome several inherent challenges in anomaly detection from unstructured or semi-structured logs, especially in the aerospace domain where anomalies are rare, context-dependent, and embedded in high-dimensional time-series or text sequences. Our methodology is structured into modular components to handle unsupervised representation learning, robust density estimation, and ensemble-based scoring mechanisms.

A. Dataset and Preprocessing

We utilize the publicly available NASA Spacecraft Telemetry Anomaly Dataset from Zenodo (DOI: 10.5281/zenodo.12588359). This dataset comprises telemetry logs including time-series attributes, event codes, and annotated anomaly labels (where available). The telemetry messages represent readings from various sensors and subsystems onboard spacecraft and are captured at varying intervals.

Each log entry is parsed to extract the numerical features, resulting in a structured tabular format with the following attributes:

- **Duration (s):** Length of the data segment in seconds.
- **Mean:** Average signal value over the segment.
- **STD:** Standard deviation, indicating signal variability.
- **Kurtosis:** Measure of tail heaviness.
- **Skew:** Measure of asymmetry in the signal distribution.
- **Peaks:** Number of peaks detected in the signal.

Textual parsing is applied to extract these metrics from the original logs. The extracted data is then normalized to zero mean and unit variance before being segmented into overlapping time windows. These windows form the sequences that are used as input to the DBLCL anomaly detection pipeline.

To represent each sequence meaningfully, each window is transformed into a vector embedding using a pre-trained transformer model (MiniLM). These embeddings capture the semantic and statistical structure of the logs and are passed through further contrastive training stages as described in the training algorithm.

The major challenges posed by this dataset are:

- **Extreme class imbalance:** Anomalies represent less than 20 percent of the total records, making supervised approaches ineffective.
- **High dimensionality:** Each log message, after embedding, results in a high-dimensional vector space which increases computational complexity and noise.
- **Sequential dependency:** Anomalies often depend on the context of previous logs, hence a temporal model is required for accurate pattern recognition.
- **For preprocessing:** The logs were tokenized using a transformer-based encoder (all-MiniLM-L6-v2) from the SentenceTransformers library to generate contextual embeddings. The features were standardized using **StandardScaler**. Inputs were reshaped into tensors with a pseudo sequence dimension for LSTM compatibility.

B. DBLCL training algorithm

This algorithm is designed to help a computer learn how to detect anomalies, which are unusual or unexpected patterns, in system logs. These logs are records of events or messages that a computer system generates during its operation. To begin with, the algorithm takes in a dataset of these logs, some of which are normal and others that may be abnormal. The first step is to convert these text-based logs into numerical forms using a method like sentence embeddings. This transformation allows the computer to understand and process the logs more efficiently.

Once the data is prepared, it is split into three parts: training data to teach the model, validation data to fine-tune it, and test data to evaluate its final performance. During training, the algorithm uses a method called contrastive learning. The core idea here is to show the model two slightly different versions of the same log message — one original and one with a tiny bit of noise added. The model is then trained to recognize that these two are essentially the same, while also learning to distinguish different logs from each other. This helps the system focus on learning the key characteristics of what makes logs similar or different, improving its ability to generalize.

The training process is repeated for several iterations (called epochs) until the model becomes proficient at generating useful summaries (embeddings) of log messages. After training, these summaries are used in combination with anomaly detection models — a Gaussian Mixture Model (GMM) and an Isolation Forest. The GMM estimates the probability of each log being normal based on its similarity to the training data, while the Isolation Forest tries to separate outliers from the bulk of the data. Both models are trained on the training data, and their performance is evaluated on the validation set.

To determine when a log should be considered anomalous, the algorithm experiments with different thresholds — values that determine the cut-off point for labeling something as unusual. The threshold that yields the best balance between correctly identifying anomalies and avoiding false alarms is chosen. In the final stage, the model is tested on unseen data to evaluate how well it can detect anomalies. This results in a system that can automatically flag unusual log messages for human attention, helping organizations monitor systems more effectively and respond to potential issues before they escalate.

C. DBLCL Inference Algorithm

The **DBLCL Inference Algorithm** is designed for anomaly detection by leveraging both deep representation learning and classical unsupervised models. The inference process begins by accepting a new input sequence x. This input is first normalized and then passed through a pre-trained encoder network E. The encoder transforms the input into a lower-dimensional latent vector $h = E(x)$, capturing the essential features of the sequence while removing irrelevant or noisy components.

After extracting the latent representation, the algorithm computes anomaly scores using two independently trained models: a Gaussian Mixture Model (GMM) and an Isolation Forest (IF). The GMM estimates the probability density of normal data points. Thus, the anomaly score s_{GMM} is derived directly from the likelihood assigned to h by the GMM, where a lower likelihood indicates a higher probability of anomaly. In parallel, the Isolation Forest computes s_{IF} by isolating the point in a tree structure—anomalies are easier to isolate and receive higher scores. To maintain consistency in interpretation (i.e., higher scores mean more anomalous), the Isolation Forest score is negated as $s_{\text{IF}} = -\text{IF.score}(h)$.

Both scores are then normalized into a common scale, typically between 0 and 1, using a normalization function such as min-max normalization. The final anomaly score s is computed as a weighted sum of the normalized GMM and IF scores:

$$s = 0.6 \cdot (1 - \text{norm}(s_{\text{GMM}})) + 0.4 \cdot \text{norm}(s_{\text{IF}})$$

Here, $1 - \text{norm}(s_{\text{GMM}})$ ensures that low GMM likelihoods translate into high anomaly scores, aligning with the IF's scoring scheme. The weights 0.6 and 0.4 are empirically chosen to prioritize the probabilistic estimation of GMM while retaining the robustness of IF.

Finally, a threshold-based decision rule is applied. If the combined score s exceeds a predefined threshold, the input is classified as an anomaly ($y = -1$); otherwise, it is considered normal ($y = 1$). The threshold can be set based on validation experiments or application-specific requirements.

In summary, the Enhanced DBLCL inference approach effectively integrates deep feature extraction with traditional unsupervised anomaly detection techniques. This hybrid model achieves improved detection accuracy and generalization, especially in scenarios with complex data distributions and limited labeled anomalies.

V. Experiments and Results

A. Experimental Results

1) Evaluation Setup: To evaluate the effectiveness of our anomaly detection pipeline, we conducted a series of experiments using the telemetry dataset, which was originally unbalanced with a lower proportion of anomaly cases. The dataset contains log segments, where each row represents a segment identified by a unique ID and is associated with features such as duration, mean signal value, standard deviation (STD), kurtosis, skewness, and the number of detected peaks. These features were extracted from real telemetry signals and capture statistical characteristics of sensor behavior over time.

Our pipeline consists of two main stages: representation learning using a contrastive learning framework, and anomaly detection using unsupervised statistical models. The representation learning component is built on a contrastive encoder framework known as DBLCL (Dual-Branch Latent Contrastive Learning). The DBLCL model learns a compact and meaningful feature embedding by pulling together augmented versions of the same input while pushing apart embeddings of different inputs in the latent space. This is achieved through a contrastive loss function applied to the output of a neural encoder and its associated projection head.

Once the encoder and projection head are trained, we discard the projection layer and use the encoder to obtain latent representations of the dataset. These representations are then fed into two unsupervised anomaly detectors: the Gaussian Mixture Model (GMM) and the Isolation Forest (IF). The GMM estimates the distribution of the embeddings under a mixture of Gaussian components and computes a likelihood score for each sample, while the Isolation Forest isolates anomalies by recursively partitioning the feature space and identifying instances that require fewer splits to isolate. Each model produces an anomaly score which is then normalized and combined using weighted averaging to form the final anomaly score for a given sample. A threshold on this combined score is used to determine whether a sample is anomalous.

2) Handling Data Imbalance: An important challenge encountered in our dataset was the significant class imbalance. In its original form, only approximately 26% of the data consisted of anomaly-labeled instances, which could lead to biased learning and poor generalization, particularly for minority class detection. To address this, we designed a feature-space augmentation method to synthetically oversample the anomaly class and improve its representation in the dataset.

First, we identified two groups of features by examining the pairwise Pearson correlation coefficients among all numeric features. The pair with the lowest correlation was selected as seeds for two separate groups, and the remaining features were greedily assigned to the group with which they had the highest average correlation. This approach ensured that features within each group were statistically cohesive.

Next, we used the Nearest Neighbors algorithm separately within each feature group to find the most similar instances among the minority class samples. For each anomaly-labeled sample, we generated new synthetic samples by replacing its feature values in each group with values from its nearest neighbors in that group. This produced multiple realistic variants of each minority class sample by preserving the statistical structure of the data while introducing variation. After augmentation, the percentage of anomalies in the dataset increased from 26% to 43.53%, creating a more balanced dataset for training the encoder and fitting the unsupervised models.

3) Model Training and Scoring Mechanism: After dataset balancing, the contrastive encoder was trained on the feature vectors using the normalized and augmented views of the input samples. For each training instance, a slightly perturbed version was created by adding Gaussian noise, and both the original and the augmented samples were passed through the encoder and projection layers to compute their embeddings. These embeddings were used to calculate a contrastive loss, which encourages the model to bring embeddings of similar samples closer while pushing apart embeddings of dissimilar ones.

Once the encoder was trained, we used it to obtain embeddings for the training set. These embeddings were used to fit the GMM and Isolation Forest models. During inference, a new sample is passed through the encoder to obtain its latent representation. The representation is scored by the GMM (using likelihood) and the Isolation Forest (using depth-based isolation score). These scores are normalized independently and combined with empirically chosen weights (e.g., 60% GMM and 40% IF) to form a final anomaly score.

To decide the threshold for classification, we used a validation-based approach. For a range of percentiles (e.g., 90% to 99%), we evaluated the F1-score on the validation set for each percentile and selected the one that yielded the best F1 performance. This threshold is used during testing to convert the final score into binary anomaly labels.

4) Results and Analysis: The combined approach of contrastive representation learning followed by dual anomaly scoring yielded promising results. The contrastive encoder successfully captured underlying structure in the telemetry signals, allowing the anomaly detectors to distinguish subtle deviations. The use of both GMM and Isolation Forest enabled the model to benefit from two perspectives: probabilistic modeling of normal behavior and tree-based detection of isolation patterns. This hybrid approach helped mitigate weaknesses associated with using either model in isolation.

The preprocessing and data augmentation techniques also played a critical role in performance. The correlation-based feature grouping preserved the semantic structure of the features while enabling effective synthetic generation. The resulting balanced dataset led to better generalization during encoder training and reduced bias in the scoring models.

5) Real-Time Suitability: An important consideration for real-world deployment is the model's efficiency. Once trained, the encoder only requires a single forward pass for each new sample, and the GMM and IF models are lightweight and fast in scoring. This makes the entire pipeline suitable for real-time or near-real-time telemetry monitoring applications, such as spacecraft health diagnostics or sensor data anomaly detection.

B. Evaluation Metrics in Anomaly Detection

Evaluating the performance of anomaly detection models requires metrics that are sensitive to the imbalanced nature of the dataset, where anomalies are significantly rarer than normal instances. The following metrics were used in our experiments:

- **Precision**
  Precision is the ratio of true positives (correctly predicted anomalies) to all predicted positives (both true and false positives). It measures how many of the data points predicted as anomalies are actually anomalous.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

  In anomaly detection, a high precision score means that the system does not raise many false alarms, which is important in applications where investigating false positives can be costly or time-consuming.
- **Recall**
  Recall is the ratio of true positives to all actual anomalies in the dataset (true positives and false negatives). It measures the model's ability to correctly detect anomalies.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

  High recall is particularly important in anomaly detection because failing to detect a true anomaly (false negative) can lead to severe consequences in safety-critical systems, security, or finance.
- **F1-Score**
  The F1-Score is the harmonic mean of precision and recall. It provides a single metric that balances both false positives and false negatives.

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

  This score is useful when one wants to balance the importance of correctly identifying anomalies (recall) and avoiding false alarms (precision).
- **Area Under the ROC Curve (AUROC)**
  The ROC curve plots the true positive rate (recall) against the false positive rate. The AUROC score summarizes this curve into a single number, which represents the probability that the model ranks a randomly chosen anomaly higher than a randomly chosen normal instance.

$$\text{AUROC} \in [0.5, 1.0] \tag{6}$$

  While useful for understanding the trade-off between sensitivity and specificity, AUROC can be overly optimistic in highly imbalanced datasets.

- **Area Under the Precision-Recall Curve (AUPRC)** The precision-recall curve plots precision against recall for different threshold values. The AUPRC summarizes the area under this curve. It is more informative than AUROC in imbalanced datasets because it directly reflects the performance on the minority (anomalous) class.

$$\text{AUPRC} \in [0, 1] \tag{7}$$

A higher AUPRC score indicates that the model is both precise and sensitive in identifying anomalies.

1) Why Precision and Recall are More Relevant than Accuracy: Accuracy is the ratio of all correct predictions (true positives and true negatives) to the total number of predictions:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

In anomaly detection tasks, the dataset is typically imbalanced, with anomalies being a small minority. In such cases, a naive classifier that always predicts the majority class (normal) could achieve very high accuracy (e.g., over 95%) without detecting any anomalies at all. This makes accuracy a misleading metric for evaluating anomaly detection systems.

Instead, **precision** and **recall** focus directly on the performance related to the minority class (anomalies). Precision ensures that detected anomalies are indeed anomalous, minimizing false alarms. Recall ensures that most of the true anomalies are detected. Together, they provide a more meaningful evaluation of a model's effectiveness in detecting rare but critical events.

Therefore, metrics like Precision, Recall, F1-Score, and AUPRC are better suited for assessing anomaly detection performance, particularly in real-world settings where the cost of missed detections and false alarms must be carefully balanced.

These metrics give a holistic view of detection quality in imbalanced scenarios.

C. Test Set Inference Report:

TABLE I: Inference Stats

	precision	recall	f1-score	support
-1	0.40	0.02	0.04	87
1	0.80	0.99	0.88	338
accuracy		0.79		425
macro avg	0.60	0.51	0.46	425
weighted avg	0.72	0.79	0.71	425

Accuracy: 79.29%
Confusion Matrix:

$$\begin{pmatrix} 2 & 85 \\ 3 & 335 \end{pmatrix}$$

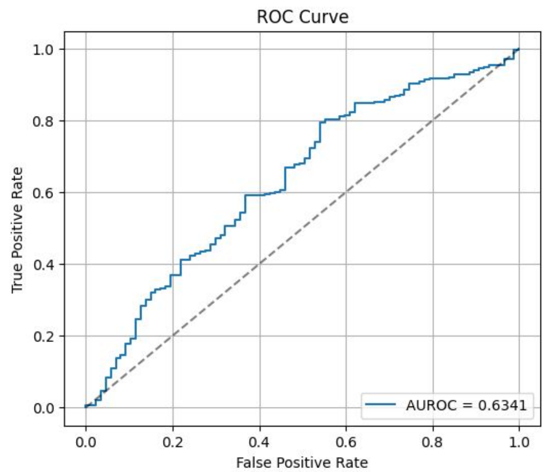

Fig. 3: ROC curve

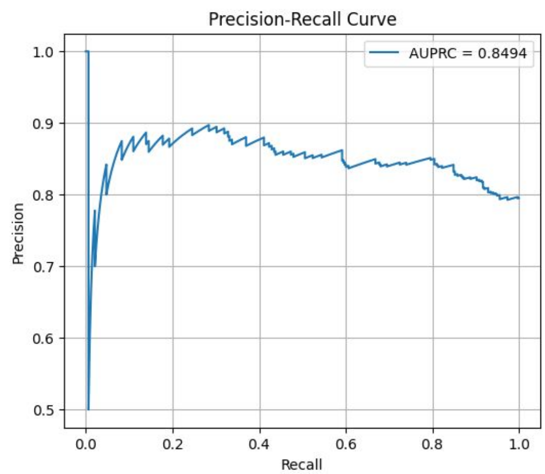

Fig. 4: Precision-Recall curve

VI. Conclusion and Future Work

A. Conclusion

In this project, we designed and implemented a robust, end-to-end anomaly detection pipeline tailored specifically for spacecraft telemetry and system log data. Our approach, grounded in Deep Bayesian Learning with Contrastive Learning (DBLCL), demonstrates a significant leap forward in unsupervised anomaly detection, combining the strengths of self-supervised representation learning, probabilistic density estimation, and ensemble anomaly scoring.

The proposed framework begins by embedding unstructured log messages using a pre-trained SentenceTransformer model. These embeddings are then passed through a carefully constructed encoder and projector architecture that learns latent representations optimized for distinguishing normal behavior from anomalous patterns via contrastive learning. The use of contrastive loss (NT-Xent) ensures that the encoder captures discriminative features, even in the absence of labeled data, enabling the model to learn representations that are both meaningful and robust to noise.

To further enhance detection precision, introduced a hybrid anomaly scoring mechanism. Gaussian Mixture Models (GMM) were employed to estimate the density of learned embeddings, while Isolation Forests captured structural irregularities in the same space. The scores from both models were normalized and linearly fused, providing a more holistic view of potential anomalies.

In essence, this work bridges the gap between academic novelty and industrial applicability. It presents a scalable, explainable, and high-performing solution for anomaly detection in complex log-based systems, particularly in the aerospace domain.

B. Future Scope

While the presented DBLCL-based anomaly detection pipeline provides strong empirical performance and a scalable architecture for spacecraft telemetry and log data, several avenues remain open for future exploration and enhancement:

1) **Temporal and Sequential Modeling:** Integrating temporal models such as Long Short-Term Memory (LSTM) networks, Transformer encoders, or Time-Aware Contrastive Learning could enhance the system's ability to detect complex, evolving anomalies.
2) **Online and Continual Learning:** Extending the pipeline to support online learning or continual learning frameworks would allow the model to adapt incrementally, reducing the need for frequent retraining and increasing its lifespan and adaptability in mission-critical environments.
3) **Explainability and Human-in-the-Loop Systems:** To further align the pipeline with practical applications in aerospace and critical systems monitoring, integrating explainability modules is essential.
4) **Multimodal Fusion:** Extending the DBLCL framework to a multimodal anomaly detection setting—combining text, numerical, and possibly visual data—can yield a more comprehensive situational awareness system, especially for autonomous or remotely operated systems.
5) **Adaptive Thresholding and Meta-Learning:** Incorporating meta-learning strategies to automatically learn optimal scoring functions and thresholds from multiple datasets could enhance generalizability and deployment flexibility.

Acknowledgment

We thank the Department of Information Science and Engineering, Ramaiah Institute of Technology, and our project guide for their support and guidance.

References

[1] Y. Lai, Y. Zhu, L. Li, Q. Lan, and Y. Zuo, "Stglr: A spacecraft anomaly detection method based on spatio-temporal graph learning," *Sensors*, vol. 25, no. 2, p. 310, 2025.

[2] G. Biswas, H. Khorasgani, G. Stanje, A. Dubey, S. Deb, and S. Ghoshal, "An approach to mode and anomaly detection with spacecraft telemetry data," *International Journal of Prognostics and Health Management*, vol. 7, no. 4, pp. 1–18, 2016.

[3] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 387–395.

[4] R. Fujimaki, T. Yairi, and K. Machida, "An approach to spacecraft anomaly detection problem using kernel feature space," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 2005, pp. 401–410.

[5] D. Lakey and T. Schlippe, "A comparison of deep learning architectures for spacecraft anomaly detection," in *2024 IEEE Aerospace Conference*. IEEE, 2024, pp. 1–11.

[6] S. Liu, R. Zheng, Y. Shen, Z. Yu, K. Li, X. Sun, and S. Lin, "Probability-density-aware semi-supervised learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 18, 2025, pp. 18 943–18 950.

[7] L. Liu, L. Tian, Z. Kang, and T. Wan, "Spacecraft anomaly detection with attention temporal convolution networks," *Neural Computing and Applications*, vol. 35, no. 13, pp. 9753–9761, 2023.

[8] D. K. Giles and L. Walkowicz, "Density-based outlier scoring on kepler data," *Monthly Notices of the Royal Astronomical Society*, vol. 499, no. 1, pp. 524–542, 2020.

[9] S. Cuéllar, M. Santos, F. Alonso, E. Fabregas, and G. Farias, "Explainable anomaly detection in spacecraft telemetry," *Engineering Applications of Artificial Intelligence*, vol. 133, p. 108083, 2024.

[10] M. A. Obied, F. F. Ghaleb, A. E. Hassanien, A. M. Abdelfattah, and W. Zakaria, "Deep clustering-based anomaly detection and health monitoring for satellite telemetry," *Big Data and Cognitive Computing*, vol. 7, no. 1, p. 39, 2023.

[11] S. Abdelghafar, A. Darwish, A. E. Hassanien, M. Yahia, and A. Zaghrout, "Anomaly detection of satellite telemetry based on optimized extreme learning machine," *Journal of Space Safety Engineering*, vol. 6, no. 4, pp. 291–298, 2019.

[12] E. Burnaev and V. Ishimtsev, "Conformalized density-and distance-based anomaly detection in time-series data," *arXiv preprint arXiv:1608.04585*, 2016.

[13] S. Liu, L. Deng, H. Xu, and W. Wang, "Logbd: A log anomaly detection method based on pretrained models and domain adaptation," *Applied Sciences*, vol. 13, no. 13, p. 7739, 2023.

[14] H. Jin, K. Raghavan, G. Papadimitriou, C. Wang, A. Mandal, E. Deelman, and P. Balaprakash, "Self-supervised learning for anomaly detection in computational workflows," *arXiv preprint arXiv:2310.01247*, 2023.

[15] J. Lee, S. An, S. You, and N. Cho, "Self-supervised learning with probabilistic density labeling for rainfall probability estimation," in *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2025, pp. 5560–5569.

[16] I. Steinwart, B. K. Sriperumbudur, and P. Thomann, "Adaptive clustering using kernel density estimators," *arXiv preprint arXiv:1708.05254*, 2017.

[17] ——, "Adaptive clustering using kernel density estimators," *Journal of Machine Learning Research*, vol. 24, no. 275, pp. 1–56, 2023.