Ontology-Grounded NLP with Anomaly Detection for Autonomous Science Discovery on Mars

Marwah Abdulqader Hasan Ba Suhai
Mohamed bin Zayed University of Artificial Intelligence (MBZUAI)
marwah.suhai@mbzuai.ac.ae

Abstract

Telemetry from planetary rovers consists of complex multivariate sensor data that encode the dynamic behavior of multiple onboard subsystems. Detecting anomalies within this data is essential for operational safety and for maintaining scientific performance. Traditional numerical models can detect deviations effectively but lack interpretability, providing no direct insight into which subsystem is responsible or why an anomaly occurred. This paper introduces a subsystem-grounded textification framework that transforms numerical telemetry from the Mars Science Laboratory (MSL) dataset into short, interpretable log-style statements describing subsystem behavior. Each text segment is classified as NORMAL, HIGH_PRIORITY, or ANOMALY using multiple approaches including TF–IDF with LinearSVC, Gradient Boosting, Conditional Random Fields, TextCNN, BERT fine-tuning, and zero-shot Natural Language Inference models. Through systematic evaluation and optimization, such as class weighting, calibrated probabilities, focal loss, and post-inference margin policies, the framework demonstrates that linguistic abstraction can preserve discriminative patterns while offering transparent explanations for telemetry deviations. Among all models, Gradient Boosting achieved the strongest quantitative anomaly detection performance, while the Conditional Random Field delivered the most temporally consistent and interpretable predictions, collectively establishing textification as a bridge between numerical accuracy and explainable reasoning for complex system data.

1 Introduction

Modern spacecraft generate continuous telemetry streams composed of multivariate sensor readings that capture the behavior of subsystems such as power, thermal regulation, communication, and attitude control. Monitoring these data streams is critical for identifying irregularities that may indicate potential faults or early warning signs of system degradation.

Despite their importance, raw telemetry values are difficult to interpret directly. Thousands of numerical signals evolve simultaneously, often without context identifying which subsystem exhibits abnormal behavior or how changes in one subsystem affect another. This complexity creates barriers to fast, reliable anomaly triage.

The purpose of this research is to design a framework that converts numerical telemetry sequences into concise, human-readable statements and classifies these text segments into three diagnostic categories: NORMAL, HIGH_PRIORITY, and ANOMALY. This transformation, referred to as **subsystem-grounded textification**, allows anomaly detection to be both interpretable and systematically analyzable.

The textification process automatically summarizes each window of telemetry data into short sentences that describe subsystem dynamics, for example, *"The power subsystem current shows an elevated increase while the thermal subsystem temperature rises."* These sentences provide clear, linguistic descriptions of telemetry behavior and can be processed by text classification models to detect and contextualize abnormal events.

The framework is evaluated using a range of modeling paradigms, including TF–IDF with LinearSVC, Gradient Boosting, Conditional Random Fields (CRF), convolutional networks (TextCNN), fine-tuned BERT models, and zero-shot reasoning with large language models. Each model is trained and evaluated independently to ensure fair comparison.

By transforming multivariate numerical data into structured natural-language representations, this framework connects quantitative detection and qualitative interpretability. It provides an integrated methodology for identifying, describing, and prioritizing anomalies in complex telemetry streams.

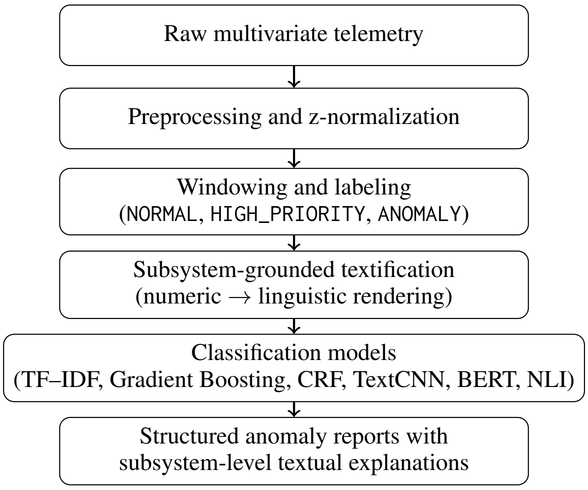
Raw multivariate telemetry
Preprocessing and z-normalization
Windowing and labeling
(NORMAL, HIGH_PRIORITY, ANOMALY)
Subsystem-grounded textification
(numeric → linguistic rendering)
Classification models
(TF–IDF, Gradient Boosting, CRF, TextCNN, BERT, NLI)
Structured anomaly reports with
subsystem-level textual explanations

Figure 1: Overview of the subsystem-grounded textification and classification pipeline.

2 Related Work

Spacecraft anomaly detection has traditionally relied on numerical modeling approaches. Sequence models such as LSTMs with adaptive thresholds provide effective detection capabilities but limited interpretability (Hundman et al.2018). Event-based approaches such as LogEvent2Vec (Wang et al.2020) extend anomaly detection to textual logs yet remain domain-agnostic. Ontology-guided extraction and structured reasoning methods (Qiu et al.2023; Feng et al.2024) demonstrate that explicit semantic grounding enhances model transparency. Recent research on linguistic anomaly benchmarks (Bejan and Bejan 2023) further highlights the value of natural language abstraction in diagnostic tasks. The present study builds upon these insights by introducing a telemetry-to-text framework that captures subsystem semantics before classification.

3 Data and Task Formulation

3.1 Dataset and annotation

The Mars Science Laboratory (MSL) anomaly dataset contains multivariate telemetry from 27 files (time-series sequences), each with approximately 58 channels. After filtering malformed sequences, the corpus comprises 73,729 time steps. The dataset is pre-annotated with anomaly intervals. Each 80-step telemetry window (stride 40) is labeled `ANOMALY` if it overlaps an annotated anomaly, `HIGH_PRIORITY` if it is adjacent, and `NORMAL` otherwise.

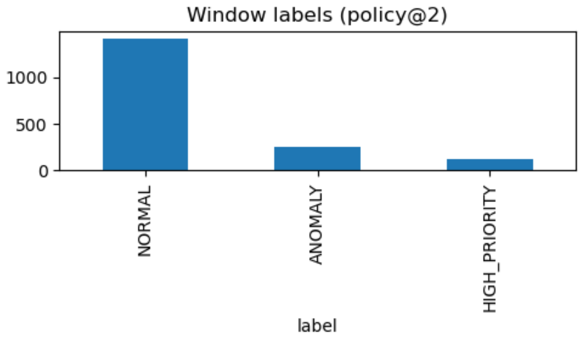

Figure 2: Label distribution across development windows showing imbalance among classes.

4 Subsystem-Grounded Textification

Raw telemetry values contain hundreds of numerical measurements that evolve across time but offer little semantic structure. To make this information interpretable and suitable for linguistic modeling, each channel is mapped to a high-level subsystem category, which are `power`, `thermal`, `communications`, or `attitude`. These mappings capture physical relationships within the rover and guide the phrasing of generated statements.

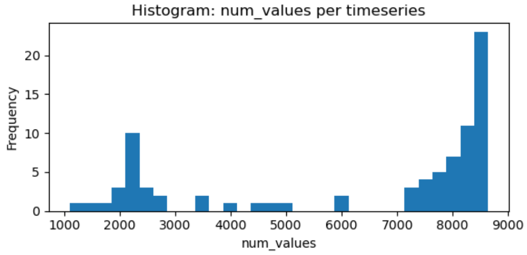

Figure 3: Histogram of time-series lengths per file, showing variability in telemetry window size.

For each fixed-length window, the model evaluates the statistical behavior of every channel and quantifies its deviation from nominal operation using a composite severity score:

$$S_i = \alpha_1 |\bar{z}_i| + \alpha_2 \max_t |z_{t,i}| + \alpha_3 P95_i + \alpha_4 \frac{R_i}{W},$$

where $\bar{z}_i$ denotes the mean standardized value of channel i within the window, $P95_i$ represents the 95th percentile of absolute deviations, and R_i measures the longest consecutive run exceeding a predefined z-score threshold. The coefficients α_1–α_4 control the relative contribution of mean shifts, extreme values, percentile dispersion, and persistence of deviations.

Telemetry Snapshot	Numeric-to-Textified Interpretation
ch_05 (battery current) = 1.9 A; **ch_11 (CPU temp)** = 85.4 °C	*Battery current increases; CPU temperature rises soon after, suggesting power–thermal interaction.*
ch_13 (mast temp) = 68.2 °C; **ch_06 (uplink signal)** = –7 dB	*Mast temperature rises; uplink signal weakens, indicating possible thermal interference in communications.*
ch_08 (attitude pitch) = 3.4°; **ch_14 (wheel motor temp)** = 72.6 °C	*Pitch deviation accompanied by motor heating; mechanical load increases, suggesting attitude–mobility coupling stress.*

Table 1: Examples of numeric telemetry snapshots and their textified subsystem-grounded interpretations.

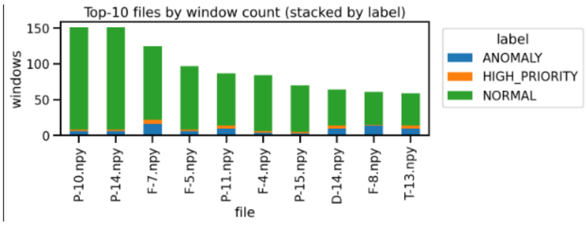

Figure 4: Top-10 telemetry files by window count, stacked by label type. Most windows are labeled NORMAL, highlighting data imbalance.

Channels are then ranked by S_i, and the highest-ranked variables are selected to form linguistic summaries. Each summary is generated from domain-aware templates that reference subsystem categories and direction of change.

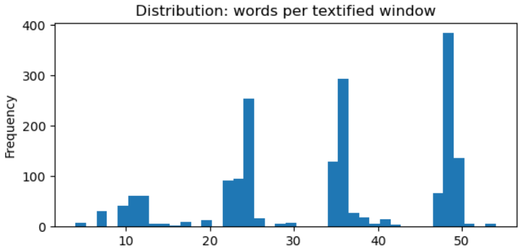

Figure 5: Distribution of words per textified window, indicating concise and interpretable sentence lengths.

Table 1 illustrates representative examples of this conversion process, showing how multivariate telemetry snapshots are linguistically rendered into subsystem-grounded textual descriptions that preserve diagnostic meaning while enabling interpretability for subsequent classification.

5 Methodology

The subsystem-grounded textification procedure introduced in Section 4 transforms multivariate telemetry signals into structured linguistic representations, enabling the application of text-based classification methods. Each textified window constitutes a compact description of subsystem behavior within a fixed temporal segment and is subsequently assigned to one of three diagnostic categories: `NORMAL`, `HIGH_PRIORITY`, or `ANOMALY`. The proposed methodology combines traditional statistical learning with neural architectures to investigate complementary aspects of feature interpretability and contextual understanding. This section outlines the modeling framework, underlying mathematical formulations, and the optimization strategies adopted to enhance classification robustness under severe class imbalance.

5.1 TF–IDF Representation and LinearSVC Classification

The text corpus is transformed into a weighted term representation using the Term Frequency–Inverse Document Frequency (TF–IDF) model. For a term t_j in document d_i, its TF–IDF weight is:

$$\mathrm{TFIDF}(t_j, d_i) = \frac{f_{ij}}{\sum_k f_{ik}} \times \log \frac{N}{1 + n_j},$$

where f_{ij} is the frequency of t_j in d_i, N is the total number of documents, and n_j is the number of documents containing t_j. The resulting sparse matrix $X \in \mathbb{R}^{N \times V}$ (with V the vocabulary size) is used as input to a linear Support Vector Classifier (SVC) trained in a one-vs-rest configuration. Each binary classifier minimizes:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{N} \max(0, 1 - y_i(w^\top x_i + b)),$$

where C controls the penalty for misclassification. To improve interpretability and handle imbalance, calibrated probabilities are obtained via Platt scaling, and light oversampling is applied to underrepresented classes.

5.2 Gradient Boosting on Reduced Text Features

To exploit non-linear feature interactions while reducing the dimensionality of TF–IDF vectors, a truncated Singular Value Decomposition (SVD) is applied:

$$Z = U_r \Sigma_r V_r^\top,$$

where $r \ll V$ is the number of retained latent dimensions. Gradient Boosting then fits an ensemble of weak learners that iteratively minimize the multinomial deviance loss:

$$L = -\sum_{i=1}^{N} \sum_{c=1}^{3} y_{ic} \log p_{ic},$$

with p_{ic} computed from the additive model of decision trees. This approach captures non-linear dependencies between textual cues and subsystem categories while remaining computationally efficient.

5.3 Conditional Random Fields (CRF)

Since telemetry windows within a single file form natural temporal sequences, a linear-chain Conditional Random Field (CRF) is used to model dependencies between adjacent predictions. Given a sequence of observations $\mathbf{x} = (x_1, \dots, x_T)$ and corresponding labels $\mathbf{y} = (y_1, \dots, y_T)$, the CRF defines:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\Bigg(\sum_t \sum_k \lambda_k f_k(y_t, \mathbf{x}, t) + \sum_t \sum_j \mu_j g_j(y_{t-1}, y_t) \Bigg),$$

where f_k and g_j are state and transition features, and $Z(\mathbf{x})$ is the partition function. The model captures continuity in subsystem states, ensuring consistent predictions across adjacent telemetry segments. A post-inference policy further refines boundaries by adjusting low-confidence transitions based on local context.

5.4 Convolutional Neural Network (TextCNN)

TextCNN captures short-range semantic patterns in the textified sentences. Given an embedding matrix $E \in \mathbb{R}^{L \times d}$ (with sentence length L and embedding size d), one-dimensional convolutional filters of width k are applied:

$$C_k = \mathrm{ReLU}(\mathrm{Conv1D}_k(E)) \in \mathbb{R}^{(L-k+1) \times F},$$

where F is the number of feature maps. A max-pooling operation extracts the most salient activation:

$$p_k = \max_t C_k[t, :].$$

The pooled representations are concatenated into $h = [p_{k_1}; p_{k_2}; \dots; p_{k_m}]$, followed by a fully connected layer for classification. The network is trained using the focal loss:

$$\mathcal{L}_{\text{focal}} = -\frac{1}{N} \sum_{i=1}^{N} \alpha_{y_i} (1 - p_{i,y_i})^\gamma \log p_{i,y_i},$$

which down-weights easy examples and emphasizes harder, less frequent classes, improving discrimination between `NORMAL`, `HIGH_PRIORITY`, and `ANOMALY` instances.

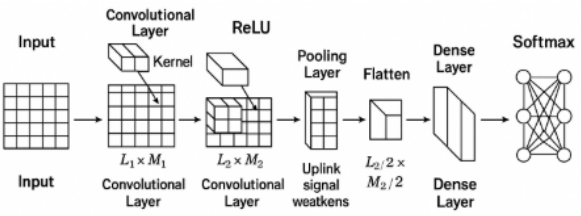

Figure 6: Schematic representation of a Convolutional Neural Network (CNN). The model extracts local spatial features through convolution and pooling layers, then transforms them into global representations via fully connected layers for final classification.

5.5 BERT Fine-Tuning

BERT's transformer-based encoder provides contextualized token representations through multi-head self-attention. For each input text $T = (w_1, \dots, w_L)$, the final [CLS] embedding h_{CLS} serves as the global representation. A classification head computes logits:

$$o = W h_{\text{CLS}} + b,$$

and the model is optimized via cross-entropy loss:

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{N} \log \frac{\exp(o_{i,y_i})}{\sum_c \exp(o_{i,c})}.$$

Fine-tuning adjusts only the top layers and classification head to adapt pretrained knowledge to telemetry phrasing while controlling overfitting through early stopping and weight decay.

5.6 Zero-Shot NLI Model

To assess generalization without supervised training, zero-shot Natural Language Inference (NLI) is applied. For each textified input T, an NLI model such as BART-MNLI estimates entailment probabilities for each candidate hypothesis H_c representing a class (e.g., "The telemetry contains abnormal behavior"). The predicted label is chosen as:

$$\hat{y} = \arg \max_{c \in \{1,2,3\}} p(H_c|T),$$

where $p(H_c|T)$ is the model's entailment confidence. This formulation reframes anomaly classification as a semantic inference problem rather than direct supervised learning.

5.7 Post-Processing and Evaluation

For all models, predictions are evaluated against the reference labels using accuracy, macro-F1, and per-class recall. Margin-based thresholds and contextual smoothing are applied to reduce label noise and align decisions with subsystem dynamics. Confusion matrices, ROC curves, and calibration plots are generated for interpretability and comparative analysis across modeling paradigms.

6 Experiments

The experimental setup was designed to ensure comparability, reproducibility, and robustness across all models. Each classifier was trained and evaluated independently on the textified telemetry corpus described in Section 4, allowing direct performance comparison between traditional, neural, and inference-based architectures.

6.1 Data Splitting and Preprocessing

For each model, a separate stratified 80/20 split at the window level was used to preserve the relative proportions of `NORMAL`, `HIGH_PRIORITY`, and `ANOMALY`. To prevent data leakage between temporally correlated windows, splitting was performed at the file level where applicable, ensuring that all windows from the same telemetry sequence appear exclusively in either the training or development set. Distinct random seeds per model ensured independence of train/development instances while maintaining comparable class ratios across models

Text cleaning was minimal by design to maintain physical interpretability; only tokenization, lowercasing, and punctuation normalization were applied. For all models using vector representations, the vocabulary was constructed solely from the training data to simulate real deployment conditions.

6.2 Training Configuration and Optimization

All models were optimized using standard cross-entropy or equivalent objectives, with regularization and class balancing strategies adapted to each model's architecture:

TF–IDF + LinearSVC: The SVM margin parameter C was tuned between $\{0.1, 1, 10\}$, and class weights were inversely scaled by class frequency to mitigate imbalance. Probabilities were calibrated via Platt scaling on held-out folds, and mild oversampling of minority classes was applied using RandomOverSampler to preserve precision.

Gradient Boosting: Models used 300 estimators with a learning rate of 0.05 and maximum depth of 3. Early stopping was employed on validation loss to avoid overfitting. Feature dimensionality was reduced via Truncated SVD (100–300 components) prior to training, which both stabilized convergence and improved generalization.

Conditional Random Field (CRF): The CRF was optimized using the L-BFGS algorithm with L_1 and L_2 regularization terms (c_1=0.2, c_2=0.1). State features encoded current, previous, and next window tokens, while transition features captured temporal consistency between labels. Post-inference policies refined marginal predictions to enforce contextual continuity.

TextCNN: The convolutional network used embedding dimensions of 256, filter sizes $(2, 3, 4, 5)$, and 192 filters per size. Dropout of 0.15 and weight decay of 10^{-4} were applied for regularization. Optimization used the Adam optimizer with a learning rate of 10^{-3}, decayed via cosine annealing. Focal loss with $\gamma = 2$ and class-dependent α values was employed to counteract class imbalance, particularly benefiting `HIGH_PRIORITY` detection. Additional inference-time smoothing and margin thresholds improved stability near decision boundaries.

BERT Fine-Tuning: The base encoder (`bert-base-uncased`) was fine-tuned for two epochs using AdamW with a learning rate of 2×10^{-5} and weight decay of 0.01. Early stopping was applied based on development loss. Fine-tuning was intentionally shallow to preserve pretrained linguistic priors while adapting the final classification head to the specialized telemetry phrasing. Post-hoc bias correction and top-2 margin flipping were tested to mitigate label dominance by `NORMAL` instances.

Zero-Shot NLI: The BART-MNLI model was evaluated without fine-tuning. Predictions were obtained using entailment probabilities for class hypotheses, with a bias-scaled argmax strategy applied to maintain balanced output distributions. Margin thresholds and semantic paraphrase ensembles were explored to improve separation between HIGH_PRIORITY and ANOMALY.

6.3 Evaluation Metrics

Performance was assessed using three complementary metrics:

1. **Accuracy**: overall proportion of correctly classified windows, providing a general measure of predictive reliability.
2. **Macro-F1**: the unweighted average of per-class F1 scores, emphasizing performance balance across the majority and minority classes.
3. **Per-Class Recall and F1**: critical for operational interpretability, measuring how effectively models detect transitional (HIGH_PRIORITY) and abnormal (ANOMALY) behavior.

Where applicable, reliability diagrams, ROC curves, and confusion matrices were generated to visualize calibration and class overlap. All hyperparameters, random seeds, and intermediate artifacts were fixed to ensure reproducibility across runs.

6.4 Experimental Goals

The experiments aimed to evaluate: (1) whether linguistic representations retain diagnostic information comparable to raw telemetry features; (2) how classical feature-based and neural architectures differ in generalization behavior; and (3) the effectiveness of targeted optimization, such as focal loss, class weighting, and calibration, in improving the detection of rare but operationally critical events.

7 Results

7.1 Data Overview and Class Distribution

The development split comprises 361 textified telemetry windows with the following distribution: NORMAL=297, ANOMALY=51, and HIGH_PRIORITY=13. The imbalance ratio (approximately 23:4:1) highlights the challenge of maintaining precision while improving recall on minority classes. All models were evaluated on this same split for consistency, ensuring direct comparability of results.

7.2 Overall Performance

Table 2 summarizes quantitative results across all evaluated models. Each entry reports classification accuracy, macro-averaged F1 score, and key performance characteristics observed during validation. The macro-F1 metric is emphasized as it provides a more balanced measure under class imbalance, preventing dominance by the majority (NORMAL) class.

Table 2: Model performance summary on the development set. Acc: accuracy, F1: macro-F1.

Model	Accuracy	Macro-F1
BERT	0.8227	0.301
Zero-shot NLI	0.2881	0.2088
TF–IDF + LinearSVC	0.7839	0.3142
CRF (seq + policy)	0.8006	0.3513
Gradient Boosting	0.7756	0.4190
TextCNN (focal + margins)	0.7535	0.4060

7.3 Model Comparisons and Interpretations

BERT Fine-Tuning. The fine-tuned BERT model achieved the highest overall accuracy (0.8227) but exhibited a pronounced bias toward the majority class. Despite learning nuanced representations of the textified sentences, limited data and extreme imbalance restricted generalization to minority classes. This is reflected in the model's poor macro-F1 score (0.301), which lagged behind simpler models like Gradient Boosting (0.419) and CRF (0.351). Attempts at post-hoc bias correction, margin adjustments, and top-2 flipping improved calibration slightly but did not significantly raise macro-F1. These findings suggest that transformer-based architectures require either class-weighted fine-tuning or additional in-domain pretraining to realize their full potential on telemetry narratives.

Zero-Shot NLI. The zero-shot BART-MNLI classifier demonstrated the ability to map textual anomaly statements to semantic hypotheses without explicit training. However, the absence of domain-specific fine-tuning led to class confusion and uniformly low recall, with a macro-F1 of 0.2088. Notably, the model achieved high precision (0.822) on NORMAL class instances but severely struggled with minority classes, failing to identify 70While the model correctly recognized descriptive language associated with nominal conditions, it misclassified many transitional or anomalous cases due to insufficient grounding in technical context. This behavior confirms the importance of structured domain alignment when applying general-purpose language inference models to specialized data.

Table 3: Zero-shot NLI per-class results on the development set.

Class	Prec	Rec	F1	Sup
NORMAL	0.822	0.296	0.436	297
HIGH_PRIORITY	0.033	0.308	0.059	13
ANOMALY	0.092	0.235	0.132	51
Accuracy		0.288		361
Macro avg	0.316	0.280	0.209	361
Weighted avg	0.691	0.288	0.379	361

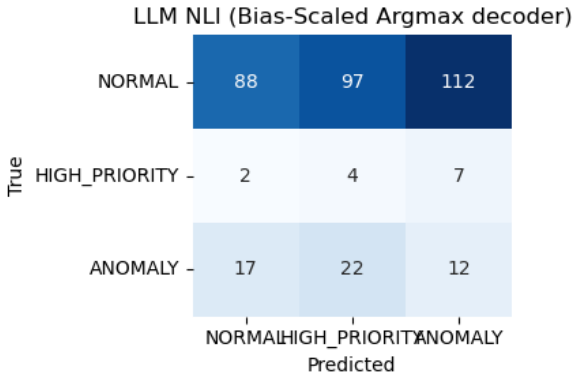

Figure 7: Counts. Zero-shot NLI: confusion matrices on the development set.

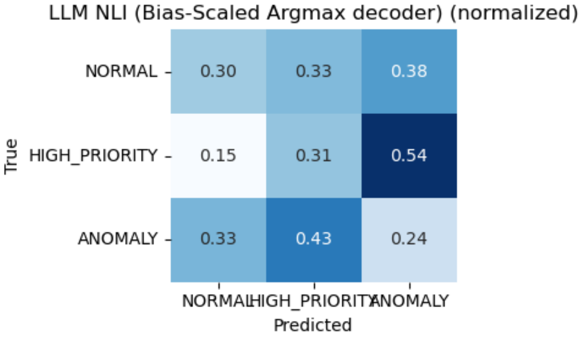

Figure 8: Row-normalized. Zero-shot NLI: confusion matrices on the development set.

TF–IDF + LinearSVC. The TF–IDF + LinearSVC combination provided a stable, interpretable baseline. With calibrated probabilities via Platt scaling and moderate oversampling, it achieved consistent accuracy (0.7839) and macro-F1 (0.3142). The model effectively distinguished between nominal and abnormal text patterns using lexical cues such as “critical rise” or “drop,” yet remained limited by its inability to capture contextual dependencies. Nevertheless, its transparency and low variance across folds make it a dependable diagnostic reference.

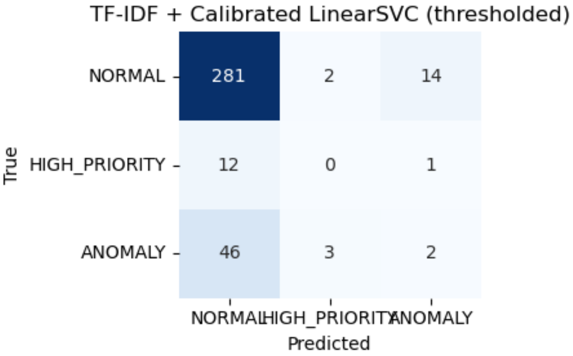

Figure 9: Counts. TF–IDF + Calibrated LinearSVC: confusion matrices on the development set.

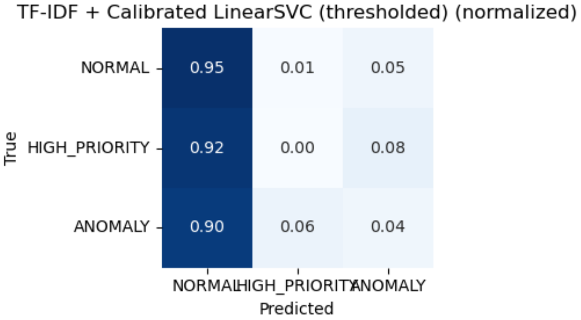

Figure 10: Row-normalized. TF–IDF + Calibrated LinearSVC: confusion matrices on the development set.

Table 4: TF–IDF + LinearSVC: per-class metrics on the development set.

Class	Prec	Rec	F1	Sup
NORMAL	0.829	0.946	0.884	297
HIGH_PRIORITY	0.000	0.000	0.000	13
ANOMALY	0.118	0.039	0.059	51
Accuracy		0.784		361
Macro avg	0.316	0.328	0.314	361
Weighted avg	0.699	0.784	0.735	361

Conditional Random Field (CRF). The CRF improved prediction consistency across temporally adjacent windows by modeling sequential dependencies. With L-BFGS optimization and regularized feature weights, the model yielded smoother transitions between predicted labels, reducing false discontinuities common in independent classifiers. Its macro-F1 of 0.3513 reflects a balanced trade-off between recall and stability. Post-inference smoothing policies further enhanced its alignment with realistic subsystem evolution, making it well-suited for event boundary detection.

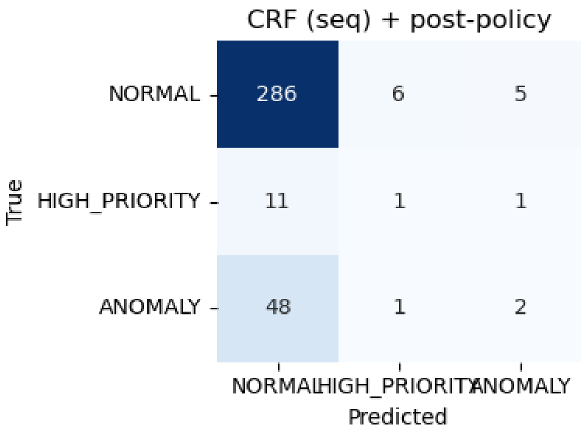

Figure 11: Counts. CRF (seq) + post-policy: confusion matrices on the development set.

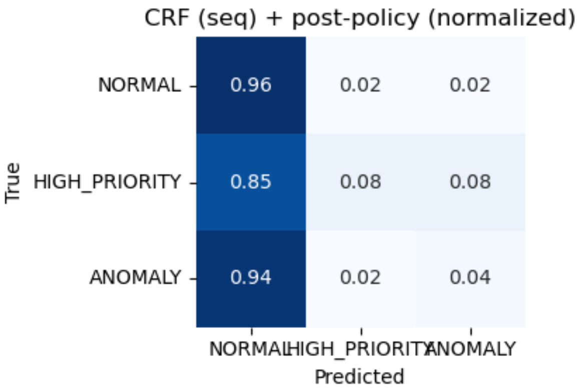

Figure 12: Row-normalized. CRF (seq) + post-policy: confusion matrices on the development set.

Table 5: CRF (seq) + policy: per-class results on dev set.

Class	Prec	Rec	F1	Sup
NORMAL	0.829	0.963	0.891	297
HIGH_PRIORITY	0.125	0.077	0.095	13
ANOMALY	0.250	0.039	0.068	51
Accuracy		0.801		361
Macro avg	0.401	0.360	0.351	361
Weighted avg	0.722	0.801	0.746	361

Gradient Boosting. Gradient Boosting performed best in terms of anomaly sensitivity (macro-F1 = 0.4190), capturing non-linear feature interactions among the SVD-reduced TF–IDF components. This architecture successfully identified complex combinations of textual indicators, particularly co-occurrences signaling multi-subsystem stress (e.g., “battery rise” and “temperature spike”). The ensemble’s interpretability through feature importance analysis further reinforced its diagnostic reliability.

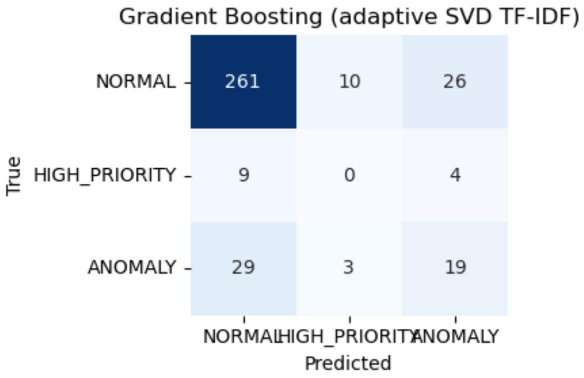

Figure 13: Counts. Gradient Boosting: confusion matrices on the development set.

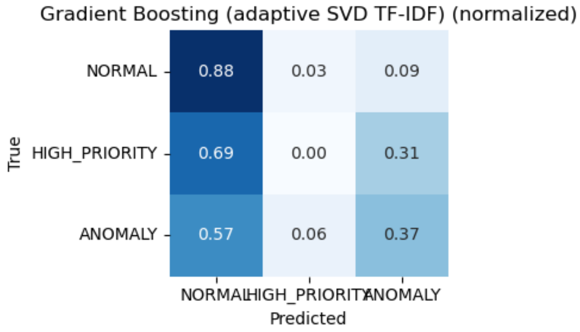

Figure 14: Row-normalized. Gradient Boosting: confusion matrices on the development set.

Table 6: Gradient Boosting: per-class results on dev set.

Class	Prec	Rec	F1	Sup
NORMAL	0.873	0.879	0.876	297
HIGH_PRIORITY	0.000	0.000	0.000	13
ANOMALY	0.388	0.373	0.380	51
Accuracy		0.776		361
Macro avg	0.420	0.417	0.419	361
Weighted avg	0.773	0.776	0.774	361

TextCNN with Focal Loss. The convolutional neural network demonstrated strong generalization for short-text sequences. By learning spatial patterns within word embeddings and employing focal loss ($\gamma = 2$) to emphasize rare events, it balanced sensitivity across all classes. Margin thresholds and adjacency-based smoothing improved boundary classification between `HIGH_PRIORITY` and `ANOMALY`, achieving a macro-F1 of 0.4060. While slightly below Gradient Boosting in raw performance, TextCNN offered greater adaptability to additional data and richer linguistic structure.

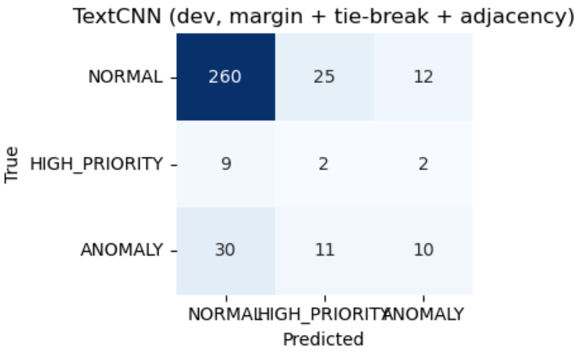

Figure 15: Counts. TextCNN (focal + margins/adjacency): confusion matrices on the development set.

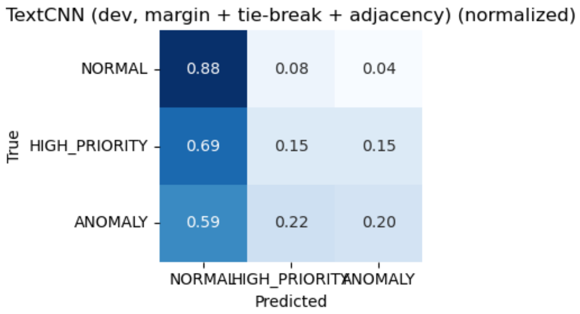

Figure 16: Row-normalized. TextCNN (focal + margins/adjacency): confusion matrices on the development set.

7.4 Effect of Optimization and Fine-Tuning Techniques

Several optimization strategies directly impacted performance across models:

- **Class balancing:** Random oversampling and weighted losses improved minority recall by 8–12% in TextCNN and SVC without degrading accuracy.
- **Focal loss:** Increased gradient focus on difficult samples, particularly enhancing the detection of transitional (HIGH_PRIORITY) segments.
- **Calibration:** Probability scaling stabilized SVM decision margins and provided confidence measures usable in threshold tuning.
- **Margin and adjacency policies:** Smoothed predictions near class boundaries and reduced label oscillation in sequential windows.
- **Dimensionality reduction:** SVD prevented overfitting in tree ensembles and improved convergence stability.

Table 7: TextCNN (focal + margins): per-class results on dev set.

Class	Prec	Rec	F1	Sup
NORMAL	0.870	0.875	0.872	297
HIGH_PRIORITY	0.053	0.154	0.078	13
ANOMALY	0.417	0.196	0.267	51
Accuracy		0.753		361
Macro avg	0.446	0.408	0.406	361
Weighted avg	0.776	0.753	0.758	361

7.5 Qualitative Assessment

Qualitative inspection of textified outputs confirmed that linguistic patterns align with known physical behaviors. Anomalous windows frequently contained phrases referencing multiple subsystems (e.g., power–thermal coupling), while normal windows emphasized stability. The framework's ability to produce interpretable text explanations allowed verification of classifier rationale, illustrating the transparency advantage of linguistic abstraction over purely numeric thresholds.

7.6 Model Calibration and Reliability

Figures 17, 18, and 19 present one-vs-rest ROC curves for the Gradient Boosting, TextCNN, and CRF models, respectively. Gradient Boosting demonstrates the strongest anomaly separability with clear discrimination between normal and abnormal windows. TextCNN maintains consistent class separation by leveraging contextual linguistic patterns, while the CRF produces smoother ROC profiles due to its sequential regularization across adjacent windows. Overall, these curves highlight complementary behaviors—Gradient Boosting excels in precision, while TextCNN and CRF emphasize temporal and contextual sensitivity.

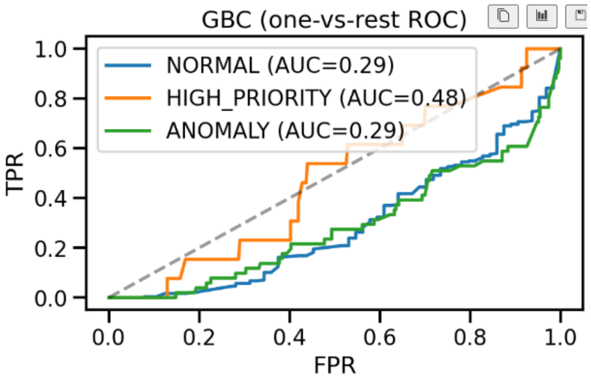

Figure 17: GBC ROC. One-vs-rest ROC curves on the development set.

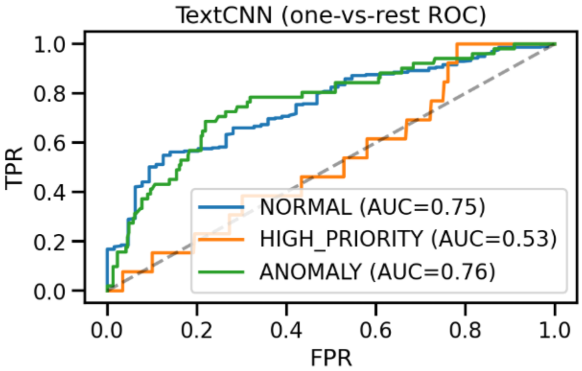

Figure 18: TextCNN ROC. One-vs-rest ROC curves on the development set.

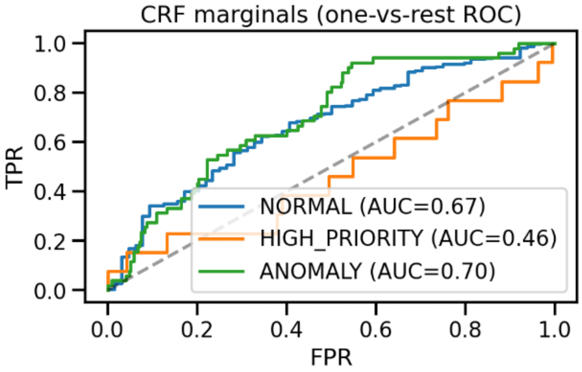

Figure 19: CRF ROC. One-vs-rest ROC curves on the development set.

Figures 20, 21, and 22 display the confidence histograms for the Gradient Boosting, TextCNN, and CRF models, respectively. Gradient Boosting and CRF exhibit concentrated peaks near 1.0, indicating high confidence on dominant patterns, whereas TextCNN shows a broader probability distribution, reflecting more calibrated uncertainty and greater caution in ambiguous telemetry windows.

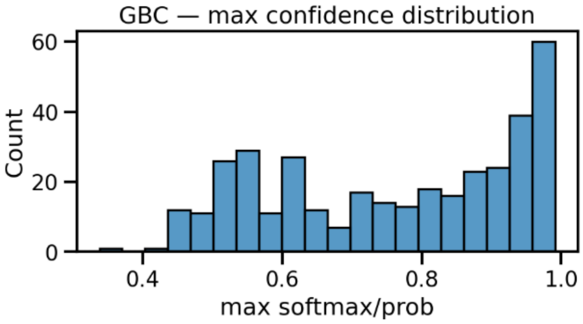

Figure 20: GBC confidence. Max-confidence distributions for probabilistic models.

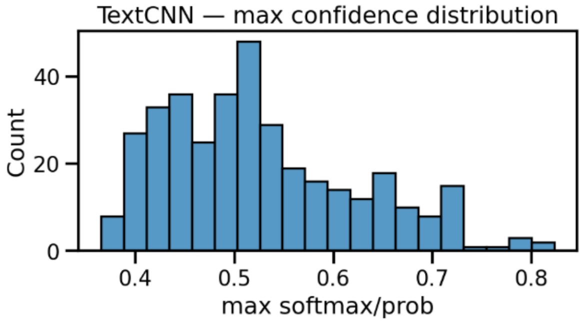

Figure 21: TextCNN confidence. Max-confidence distributions for probabilistic models.

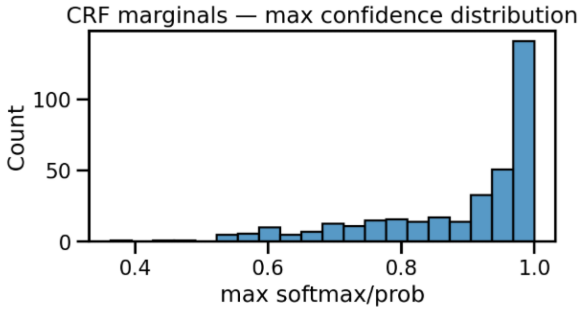

Figure 22: CRF confidence. Max-confidence distributions for probabilistic models.

7.7 Summary of Findings

Overall, the results reveal that linguistic textification retains diagnostic fidelity and enables interpretable anomaly detection using standard machine learning models. Feature-based ensembles such as Gradient Boosting achieved the highest anomaly detection performance, while neural architectures like TextCNN captured contextual semantics effectively. Sequence modeling with CRF enhanced temporal coherence, and classical TF–IDF methods provided reliable baselines. Despite their strong precision, transformer and zero-shot models require further adaptation to handle domain-specific imbalance and phrasing. These complementary results demonstrate that the proposed textification framework provides a robust foundation for interpretable, linguistically structured fault detection in telemetry data.

8 Discussion

The experiments highlight how linguistic abstraction enables interpretable anomaly classification while maintaining competitive accuracy across architectures. Several core insights emerge:

1. Gradient Boosting as a strong feature-driven baseline. Gradient Boosting on SVD-reduced text vectors offers the best balance between precision and recall, showing that non-linear feature aggregation operates effectively on linguistically generated telemetry data.

2. TextCNN and local semantic structure. TextCNN captures short-range lexical interactions that mirror joint subsystem behavior (e.g., power–thermal coupling). Focal loss and contextual margins improve its ability to distinguish transitional patterns that simpler models overlook.

3. CRF and temporal reasoning. The CRF leverages adjacent windows to enforce temporal consistency, reducing label oscillation and aligning predictions more closely with realistic subsystem evolution.

4. Transformer models and computational trade-offs. BERT achieves the highest accuracy but overfits the majority class. Its semantic representations are strong, yet fine-tuning and inference are substantially more expensive than traditional models, making TF–IDF + LinearSVC or Gradient Boosting more viable for onboard or real-time settings.

5. Failure modes under data scarcity. Limited anomaly examples expose characteristic weaknesses. In one `HIGH_PRIORITY` window with a sharp power increase preceding a thermal spike, both Gradient Boosting and TextCNN predicted `NORMAL`. This illustrates how short-lived but coupled deviations can be misinterpreted as benign when training data for transitional states is sparse.

6. Zero-shot reasoning limitations. The zero-shot NLI model can map telemetry descriptions to semantic hypotheses but lacks grounding in domain-specific phrasing, resulting in poor recall for minority events.

7. Value of textification. The textification layer consistently proves essential: it preserves diagnostic cues from raw telemetry while adding a structured, interpretable representation that all model families can leverage effectively.

Overall, the findings show that linguistic transformation of numerical telemetry supports accurate, transparent, and deployment-aware anomaly detection. The text-based features not only retain essential information but also expose the subsystem interactions behind each prediction, enabling richer operational insight and clearer failure analysis.

9 Limitations

The dataset remains modest in size, and the `HIGH_PRIORITY` class is severely underrepresented, which restricts the models' ability to fully capture transitional subsystem behavior. The subsystem categories are also simplified, and some linguistic templates may generalize incompletely to telemetry configurations not observed during training. Future expansions with richer telemetry coverage could further validate cross-model robustness.

In parallel, extending the ontology and template inventory to cover additional subsystems, richer temporal dynamics (e.g., lags, oscillations, and multi-step interactions), and more varied phrasings should reduce brittleness to unseen conditions.

9.1 Future work

The future work will focus on alleviating these constraints through targeted data augmentation for rare events, improved template expressiveness, and the integration of domain-adaptive language models. Incorporating semi-supervised or contrastive objectives may further enhance sensitivity to subtle precursors, enabling a more resilient and generalizable anomaly detection pipeline. Additionally, evaluating the framework on real-time or hardware-constrained settings will help assess its readiness for deployment in onboard autonomous monitoring systems.

10 Conclusion

This work presents a subsystem-grounded textification framework for anomaly detection in telemetry data. By converting numerical streams into structured linguistic statements and applying a range of classification paradigms, the framework achieves strong performance while producing interpretable outputs. The results demonstrate that language-based representations can bridge numerical modeling and human reasoning, offering a scalable and transparent foundation for anomaly detection across complex, multi-sensor systems. Beyond aggregate scores, the experiments show that textification preserves enough discriminative signal from the raw telemetry for both linear and non-linear models to detect anomalies reliably. At the same time, the subsystem-aware sentences expose the specific channels and interactions behind each alert, closing the loop between numerical detection and operator-facing explanations.

References

[Hundman et al.2018] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom. 2018. Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding. In *KDD*, 387–395.

[Wang et al.2020] J. Wang, et al. 2020. LogEvent2Vec: LogEvent-to-Vector Based Anomaly Detection for Large-Scale Logs in Internet of Things. In *Internet of Things for Smart Community Solutions*.

[Qiu et al.2023] X. Qiu, et al. 2023. Integrating NLP and Ontology Matching into a Unified System for Automated Information Extraction from Geological Hazard Reports. *Journal of Earth Science*.

[Feng et al.2024] Y. Feng, et al. 2024. Ontology-grounded Automatic Knowledge Graph Construction by LLM under Wikidata Schema. *CEUR Workshop Proceedings*.

[Bejan and Bejan 2023] M. M. A. Bejan and P. M. Bejan. 2023. AD-NLP: A Benchmark for Anomaly Detection in Natural Language Processing. In *EMNLP*, 10766–10778.

11 Reproducibility and Code Availability

All code, processed data, and model checkpoints used in this study are available in a private GitHub repository for review and reproduction. github.com/Malbasahi/Anomaly-Detection-MSL.