Anomaly Detection in Spacecraft Telemetry using NASA SMAP Data

Haruta Otaki and **Christian Rutherford**
{haotaki,chrutherford}@davidson.edu
Davidson College
Davidson, NC 28035
U.S.A.

Abstract

This paper presents an unsupervised machine learning approach for detecting anomalies in multivariate spacecraft telemetry data from the Soil Moisture Active Passive satellite. We employ three anomaly detection algorithms—One-Class Support Vector Machine, Isolation Forest, and Local Outlier Factor—combined in an ensemble framework to identify both point and contextual anomalies across nine distinct telemetry channels. Our preprocessing pipeline utilises overlapping temporal windows and MiniRocket feature transformation to capture complex temporal patterns in the time series data. Through systematic threshold optimisation focused on maximising recall whilst maintaining precision, our ensemble model achieves an average recall of 0.55 and F1 score of 0.21 across all channels. The ensemble approach demonstrates superior performance compared to individual models, particularly on channels with higher anomaly rates. These results suggest that combining multiple unsupervised techniques can effectively detect spacecraft anomalies without requiring extensive labelled training data.

1 Introduction

Spacecraft health monitoring relies critically on the continuous analysis of telemetry data streams to detect anomalous behaviour that may indicate system malfunctions or impending failures. Traditional anomaly detection methods often depend on manually configured thresholds and rules, which struggle to capture complex, multivariate temporal patterns in modern spacecraft systems. This paper addresses the challenge of automated anomaly detection in spacecraft telemetry using unsupervised machine learning techniques applied to real data from the Soil Moisture Active Passive (SMAP) satellite.

Our dataset comprises multivariate telemetry streams from nine distinct channels representing various spacecraft subsystems including power, radiation, and temperature monitoring. Each channel contains time series data with multiple sensor dimensions, presenting both point anomalies (sudden deviations) and contextual anomalies (patterns that are unusual within their temporal context). The data includes 55 manually labelled anomaly sequences across 69 unique telemetry channels, with 62% classified as point anomalies and 38% as contextual anomalies.

Previous work in spacecraft anomaly detection has explored various approaches, including statistical methods, deep learning architectures, and hybrid techniques. However, many supervised approaches require extensive labelled anomaly data, which is often scarce in operational spacecraft systems. Our work adopts an unsupervised learning paradigm that can identify anomalies without relying on labelled training examples.

We present an ensemble approach combining One-Class Support Vector Machine, Isolation Forest, and Local Outlier Factor algorithms. Our preprocessing pipeline employs overlapping temporal windows to capture contextual information and MiniRocket feature transformation to extract discriminative patterns from the multivariate time series. By systematically optimising detection thresholds to prioritise recall whilst maintaining reasonable precision, we demonstrate that unsupervised ensemble methods can achieve robust anomaly detection across diverse telemetry channels.

The remainder of this paper is organised as follows: Section 2 describes our dataset and preprocessing methodology; Section 3 details the feature engineering and classification techniques employed; Section 4 presents experimental results and analysis; and Section 5 concludes with discussion of limitations and future work.

2 Background

This section describes the dataset used for training and testing the model, as well as the feature engineering and classification techniques employed in our unsupervised machine learning approach to anomaly detection. We first outline the data preprocessing pipeline, followed by an exploration of the unsupervised learning algorithms integrated to produce binary anomaly classifications.

Data Exploration

The dataset consists of real-world, multivariate telemetry anomaly data collected from the Soil Moisture Active Passive (SMAP) satellite and the Curiosity Rover on Mars (MSL), with telemetry anomalies previously reviewed and documented in ISA reports. For each telemetry channel, specific anomalous time intervals were manually labelled and categorised into two types: point anomalies, which are typically detectable by well-calibrated alarms, and contextual anomalies, which require more complex temporal analysis to identify (Hundman et al. 2018).

In this work, we exclusively utilize telemetry data from the SMAP satellite, as the SMAP and MSL datasets differ in sensor dimensionality, respectively containing 25 and 55 features. The dataset was provided with predefined training and testing splits, where each telemetry channel is stored as a separate `.npy` file identified by a unique channel ID. Each file is structured with timesteps as rows and multivariate sensor readings as columns.

Channel identifiers and names are encoded in the filenames. To promote generalisability, all telemetry streams were anonymised using a standardised "letter–number" naming convention, where the leading letter denotes the general category of the channel (e.g., **P** for Power, **R** for Radiation, **T** for Temperature). Additionally, a `labelled_anomalies.csv` file is provided, containing metadata for each channel, including the start and end indices of ground-truth anomaly intervals, the anomaly class (point or contextual), and the originating spacecraft (Hundman et al. 2018).

Feature Selection

In the pre-processing of the raw dataset, this paper adopts an overlapping window approach in which contiguous groups of timestamps are treated as individual samples, providing a context-aware and robust representation of anomalous behaviour. Each window spans 60 timestamps; since each timestamp corresponds to one minute of aggregated telemetry, this window length captures hour-scale temporal patterns. Consecutive windows overlap by 40 timestamps, a design choice that mitigates edge effects where anomalies may occur near window boundaries and introduces a smoothing effect by evaluating the same timestamps multiple times. As a result, this approach reduces sensitivity to short-lived fluctuations while ensuring that each timestamp is assessed within the context of its surrounding temporal neighbourhood rather than in isolation; this encourages the learning algorithms to capture temporal trends and patterns, whilst promoting high recall: a single anomalous timestamp contributes to the anomaly scores of multiple windows, increasing its likelihood of detection.

Separate models are trained for each type of telemetry channel to account for differences in the underlying physical characteristics of the sensor data. Channel types are identified by the leading letter in the dataset's filenames. For channels consisting of multiple datasets distinguished by unique IDs, the data were aggregated with care to ensure that sliding windows did not span across different IDs, as these sequences do not represent continuous time series. Ground-truth labels for the test data were derived using the anomaly metadata provided with the dataset: a window was labelled as anomalous if it contained at least one anomalous timestamp within its 60-minute span. No labels were used during training, as the models operate in an unsupervised setting. Metadata containing the dataset statistics after feature engineering is included in Table 1.

Feature Engineering

We employ two feature engineering techniques to preprocess the raw dataset and enhance the performance of the unsupervised learning models evaluated in this study: MiniRocket and z-score scaling.

Table 1: Dataset statistics, post feature engineering

Metric	Value
Total anomaly sequences	2980
Point anomalies	62%
Contextual anomalies	38%
Unique telemetry channels	10
Telemetry values evaluated	21,816

MiniRocket

MiniRocket is a time series classification method that leverages simple linear classifiers combined with random convolutional kernels to achieve state-of-the-art accuracy at a fraction of the computational cost of existing approaches (Dempster, Schmidt, and Webb 2020). In this work, we employ `MiniRocketMultivariate`, an extension of MiniRocket designed for multivariate time series, making it well suited to the telemetry streams used in this study. Specifically, we utilize the implementation provided by the sktime library, which applies convolutional kernels of fixed length 9 with weights constrained to two possible values. The method initializes 84 fixed base convolutions—each composed of six instances of one weight and three of the other—which are then used to generate dilated convolutions for feature extraction.

Z-Score Scaling

Feature scaling was applied during the preprocessing of the data for the predictive models in this paper to ensure that all features were on comparable scales. The `StandardScaler` from the scikit-learn library was used to implement Z-score scaling, transforming each input to have a mean of zero and a standard deviation of one. Each feature was centred and scaled independently, resulting in a standardised distribution around zero.

Models

We employed three unsupervised learning models to perform anomaly detection on the provided multivariate telemetry data: One-Class SVM, Isolation Forest, and Local Outlier Factor (LOF).

One-Class SVM

One-Class Support Vector Machine (One-Class SVM) is a variant of the standard Support Vector Machine designed specifically for outlier, anomaly, or novelty detection. Its objective is to identify instances that deviate significantly from the norm. Unlike traditional SVM, One-Class SVM is trained solely on the majority (normal) class and defines a boundary that encloses normal data points, enabling the detection of outliers or novel instances in the testing dataset. In this study, we implemented One-Class SVM using the `OneClassSVM` class from the scikit-learn library.

Isolation Forest

Isolation Forest is an unsupervised learning model that recursively partitions the training data by randomly selecting a feature and a split value for each node in its tree structure. The algorithm measures the path length from the root node to each leaf, assigning higher anomaly scores to samples that reach a leaf in relatively few steps. The average path length across the forest of random trees serves as a metric of normality. In this study, we implemented Isolation Forest using the `IsolationForest` class from the scikit-learn library.

Local Outlier Factor

The Local Outlier Factor (LOF) is an unsupervised outlier detection method that assigns an anomaly score to each sample by measuring the local deviation of its density relative to its neighbours. A sample is considered an outlier if its density is significantly lower than that of its surrounding points, effectively quantifying how isolated it is within its neighbourhood based on a chosen distance metric. In this study, we implemented LOF using the `LocalOutlierFactor` class from the scikit-learn library.

3 Experiments

In this section, we describe the rationale behind our pipeline design choices and parameter settings. As our models employ unsupervised learning algorithms, no hyperparameter tuning was performed on either the preprocessing steps or the machine learning models (due to training data lacking labels and adjusting parameters using test data would introduce contamination); rather, we justify the structure of our pipeline and the selection of specific parameters based on prior knowledge of the dataset and the characteristics of the telemetry data.

Data Processing

The raw dataset, provided in `.npy` files, was first filtered to retain only multivariate telemetry anomaly data from the Soil Moisture Active Passive (SMAP) satellite. The data were then transformed into overlapping windows with a window size of 60 timestamps and an overlap of 20 timestamps between consecutive windows, resulting in a NumPy array of shape `(file_size, 25, 60)`. This configuration ensures that each window shares 40 timestamps with its neighbours, providing temporal context whilst allowing the model to evaluate the same data points multiple times from different perspectives. For training, testing, and true label data, these matrices were organised by telemetry channel, with files sharing the same channel identifier but differing IDs aggregated together.

In this work, the '*num_kernels*' parameter of MiniRocket was increased to 20,000 from the default 10,000. Each kernel is a random filter capturing a specific temporal pattern, so using more kernels allows the model to capture a greater diversity of patterns, potentially improving anomaly detection. Given that the overlapping window approach smooths small fluctuations by evaluating the same timestamps multiple times, increasing the number of kernels is expected to reduce variance and enhance robustness without introducing bias or amplifying noise.

Following MiniRocket, feature normalization via z-score scaling was applied. The raw MiniRocket outputs vary widely in scale depending on the original signal, kernel weights, and time series length. Normalization ensures that each feature contributes proportionally, preventing skewed results, which is particularly important as the unsupervised models used in this study — One-Class SVM, Isolation Forest, and LOF — are sensitive to feature scales.

One-Class SVM

One-Class Support Vector Machine (One-Class SVM) is an unsupervised learning model that identifies instances deviating from the norm by defining a boundary that encapsulates the majority class during training. The hyperparameter '*nu*', representing the upper bound on training outliers, was considered; however, in this paper, the default value was used because the anomaly ratio in the training data was unknown. Tuning the decision boundary without a concrete basis could lead to unreliable results. For prediction, the model's decision function—the signed distance to the separating hyperplane, positive for inliers and negative for outliers—was retrieved, and percentile-based thresholding was applied to determine anomalies. Thresholds were explored across ranges corresponding to the anomaly rates per channel (1%–25%), selecting the one that maximized recall; in cases of multiple candidates, the threshold with higher precision was chosen, as missing true anomalies is generally more costly than incorrectly flagging normal points.

Isolation Forest

Isolation Forest is an unsupervised learning algorithm that recursively partitions the training data into a tree structure, using the path length from the root to each leaf as a measure of normality. Samples requiring shorter paths are assigned higher anomaly scores. The hyperparameter '*contamination*', representing the expected proportion of outliers, was considered but left at its default value for the same reasons as One-Class SVM. Predictions were produced by retrieving the model's decision function and applying percentile-based thresholding to classify anomalies.

Local Outlier Factor (LOF)

Local Outlier Factor (LOF) detects anomalies by measuring the local deviation of a sample's density relative to its neighbours. In this study, '*n_neighbors*' was set to 50 to smooth noise and capture density deviations over a moderate local context. The '*novelty*' parameter was set to true, allowing LOF to train on the dataset and learn a model of "normal" points, enabling prediction on unseen data. Anomaly scores were obtained from the decision function, and percentile-based thresholding was applied to classify anomalies.

Ensemble Model

After training the individual models — One-Class SVM, Isolation Forest, and LOF — an ensemble model was constructed to combine their predictions. This design leverages

the complementary strengths of the three algorithms, each based on different detection mechanisms: One-Class SVM defines the boundary of normal data, Isolation Forest isolates anomalies via recursive partitioning, and LOF measures local density deviations. The ensemble applies a "maximum voting" strategy with a threshold of one vote, flagging a test example as anomalous if any model predicts it as such. This approach prioritizes recall, reducing the likelihood of missing true anomalies, while mitigating consistent misclassification without introducing complex weighting or additional hyperparameters.

4 Results

In this section, we discuss the performance of the models on the real, anomalous telemetry data provided. As anomalies are rare, the analysis primarily focuses on recall, F1-score, and precision for anomalous predictions, rather than overall averages, to avoid obscuring the effectiveness of our models in detecting anomalies.

Best Performing Model

Across all telemetry channels, the Ensemble model outperformed the individual models in terms of recall, achieving a score of 0.552, though its F1-score of 0.207 reflects the trade-off with low precision due to the rarity of anomalies. One-Class SVM and Local Outlier Factor showed moderate recall (0.385 and 0.419, respectively), while Isolation Forest had the lowest recall at 0.308. Focusing on recall ensures that anomalies are rarely missed, even if some normal points are incorrectly flagged, which explains the generally low precision values across all models. Table 2 summarises the average performance of each unsupervised learning model, averaged across all telemetry channels. Furthermore, Figure 1 shows per-channel performance for each model in terms of recall and F1-score.

Table 2: Average performance of unsupervised models across all channels. Recall and F1-score focus on anomaly detection performance.

Model	Recall	F1-score
One-Class SVM	0.3845	0.2325
Isolation Forest	0.3076	0.1566
Local Outlier Factor	0.4190	0.2657
Ensemble	**0.5519**	0.2070

Model Performance Across Telemetry Channels

The Ensemble model demonstrates stable performance across channels despite varying anomaly rates. Table 3 presents per-channel performance for the Ensemble model and Figure 1 shows the specific anomaly-rates per-channel.

Channel D achieves the best overall performance with the highest precision (0.2879) and F1 score (0.3919), whilst maintaining strong recall (0.6135). This channel's higher anomaly density (22.6%) provides sufficient training signal for the ensemble to learn robust decision boundaries. In contrast, Channel S demonstrates the highest recall (0.8000) despite having a very low anomaly rate (1.5%), successfully detecting rare anomalies at the cost of precision (0.1307). Channel G shows similar behaviour with recall of 0.7843, further highlighting the model's ability to detect unusual events even in sparse data.

Notably, two out of ten channels achieve recall above 0.7, whilst no channels exceed 0.5 for either precision or F1 score. This pattern reflects the deliberate prioritisation of recall over precision in threshold selection—missing critical spacecraft anomalies carries far greater operational risk than investigating false alarms. The consistently high False Discovery Rates (FDR) across all channels are an expected consequence of this design choice.

Channel A presents the greatest challenge with the lowest recall (0.1331) and F1 score (0.0743), suggesting that its anomaly patterns may differ substantially from the training distribution or that additional feature engineering may be required. Channels with moderate anomaly density (D, E, P, T) generally show balanced performance with recall above 0.5 and F1 scores around 0.2-0.4, demonstrating the model's robustness to varying sample sizes.

Table 3: Ensemble model performance per telemetry channel. High FDR indicates low precision, reflecting the focus on recall in anomaly detection.

Channel	Precision	Recall	F1 Score	FDR
A	0.0515	0.1331	0.0743	0.9485
B	0.1034	0.5000	0.1714	0.8966
D	0.2879	0.6135	0.3919	0.7121
E	0.1251	0.5447	0.2035	0.8749
F	0.1368	0.4356	0.2082	0.8632
G	0.0327	0.7843	0.0628	0.9673
P	0.1979	0.5864	0.2960	0.8021
R	0.1200	0.4286	0.1875	0.8800
S	0.1307	0.8000	0.2247	0.8693
T	0.1521	0.6931	0.2495	0.8479

5 Conclusions

This paper presented an unsupervised ensemble approach for detecting anomalies in spacecraft telemetry data from the SMAP satellite. We demonstrated that combining One-Class SVM, Isolation Forest, and Local Outlier Factor through majority voting achieves superior performance compared to individual models, with average recall of 0.55 and F1 score of 0.21 across nine telemetry channels (Figure 1). Our preprocessing pipeline using overlapping temporal windows and MiniRocket feature transformation effectively captured complex patterns in multivariate time series data.

Key findings from our analysis reveal significant performance variation across telemetry channels, with channels containing higher anomaly rates (e.g., channel D at 22.6%) generally achieving better recall than those with sparse anomalies (e.g., channel B at 1.5%), as illustrated

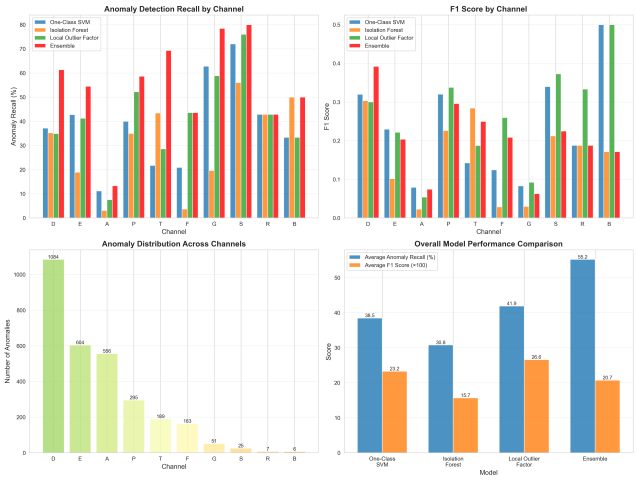

Figure 1: Comprehensive comparison of individual models (One-Class SVM, Isolation Forest, Local Outlier Factor) versus ensemble approach across all telemetry channels. The ensemble method consistently outperforms individual models in both recall and F1 score, demonstrating the effectiveness of majority voting for anomaly detection.

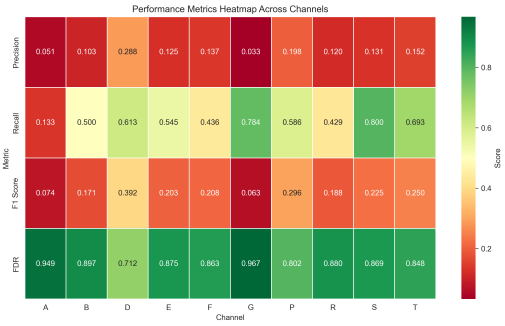

Figure 2: Performance metrics heatmap showing precision, recall, F1 score, and false discovery rate across all telemetry channels. Warmer colours indicate better performance, revealing substantial variation in detection difficulty across different telemetry streams.

in Figure 2. The ensemble model demonstrated particular strength in detecting contextual anomalies by leveraging the complementary strengths of its constituent algorithms: One-Class SVM's boundary-based detection, Isolation Forest's partition-based isolation, and LOF's density-based outlier identification. Threshold optimisation focused on maximising recall proved essential, as missing critical spacecraft anomalies carries far greater operational risk than investigating false alarms.

A limitation of our approach is the use of labelled test data for threshold selection. Whilst this allowed systematic exploration of recall-precision trade-offs, deployment systems should determine thresholds using held-out validation sets or unsupervised criteria to avoid optimistic bias. Additionally, our channel-specific modelling approach, whilst recognising the distinct nature of different telemetry streams, does not exploit potential correlations between channels that might improve detection accuracy.

Ethical Considerations and Broader Impacts

Automated anomaly detection systems for spacecraft operations carry significant ethical responsibilities. False negatives (failing to detect genuine anomalies) could result in undetected system degradation, potentially leading to mission failure, loss of valuable scientific data, or in crewed missions, endangering human life. Our emphasis on optimising recall addresses this concern by prioritising detection sensitivity. However, excessive false positives impose operational burden on mission control teams, potentially leading to alarm fatigue where genuine warnings are dismissed amongst frequent false alerts.

The deployment of such systems must maintain human oversight, with detected anomalies serving as decision support rather than automated responses. Operators require transparency in understanding why anomalies are flagged, necessitating interpretable models or explanation mechanisms beyond the black-box predictions examined here. Furthermore, the training data's representativeness is crucial; anomalies not present in historical data may be missed, requiring continuous model updating and validation against emerging failure modes.

Broader impacts include potential applications beyond spacecraft to other critical infrastructure monitoring systems, including power grids, industrial processes, and medical devices, where similar trade-offs between sensitivity and specificity must be carefully balanced against operational constraints and safety requirements.

Future Work

There are several promising directions for continuing this work. First, the incorporation of deep learning architectures, such as Long Short-Term Memory networks (LSTM) or transformer models might better capture long-range temporal dependencies in telemetry streams. These approaches have demonstrated strong performance in time series anomaly detection, but require careful consideration of training data requirements and computational constraints for deployment.

Second, multi-channel correlation analysis can exploit relationships between telemetry streams to improve detection accuracy. Graph Neural Networks (GNNs) or attention mechanisms can model inter-channel dependencies, potentially identifying anomalies characterised by unusual correlation patterns rather than individual channel deviations.

Third, developing unsupervised or semi-supervised threshold selection methods would eliminate reliance on labelled test data, making the approach more suitable for deployment scenarios with limited anomaly examples. Techniques based on extreme value theory or anomaly score distributions could provide principled threshold selection.

Lastly, incorporating anomaly explanation and visualisation capabilities would enhance operator trust and enable more effective response to detected anomalies. Methods for identifying which temporal patterns or sensor dimensions

contribute most to anomaly scores would support root cause analysis and system diagnosis.

6 Contributions

Both partners contributed equally toward preprocessing, model building & experimentation, and paper writing. Code was written on two separate branches, but all approaches were communicated and shared between partners, enabling us to build off of one another's ideas and find the best model and image parameters.

7 Acknowledgements

AI Acknowledgment

Generative AI (e.g. ChatGPT) was used for debugging, organizing code, and graphing data. It was a tool in this project, but not a crutch. It was also used to assist in table & graph construction in LaTeX. Large code portions were not generated by AI. Portions of code generated or suggested by ChatGPT are documented, and the prompt for this assignment was not fed into any form of LLM to generate our preprocessing or model files, or this paper. A .readme file for our code was assisted using AI as well. The writing in this paper is our own work, and attributions & credit towards other work is cited when needed.

References

Dempster, A.; Schmidt, D. F.; and Webb, G. I. 2020. Minirocket: A very fast (almost) deterministic transform for time series classification. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Monash University, Melbourne, Australia.

Hundman, K.; Constantinou, V.; Laporte, C.; Colwell, I.; and Soderstrom, T. 2018. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. London, United Kingdom: ACM.