Numerical Analysis of Automated Anomaly Detection Algorithms for Satellite Telemetry

Leonard Schlag*, Corey O'Meara †, Martin Wickler‡
Deutsches Zentrum für Luft- und Raumfahrt e. V., German Aerospace Center
Münchener Straße 20, 82234 Weßling, Germany

As technology evolves and the complexity of satellites and the amount of available telemetry increases, the manual inspection of thousands of parameters in detail per satellite becomes less and less manageable. While automated processes such as Out-Of-Limit (OOL) checks, which verify if a parameter exceeds an upper or lower threshold, exist, they come with the drawback of needing to be defined manually and often being very coarse to detect subtle changes in the telemetry. As this is a known problem, many space agencies are developing anomaly detection systems using machine learning methods. We found that the main difficulty in developing such an algorithm, as has been done for the Automated Telemetry Health Monitoring System (ATHMoS) at German Space Operations Center (GSOC), is minimizing the number of false positives while still detecting anomalies at a sufficiently high rate. Also, computational cost needs to be minimized since the detection algorithm needs to run at least once per day for all parameters.

Considering these important constraints specific to automatic anomaly detection for satellite telemetry, we analyse several algorithms commonly used, namely the LOF and LoOP algorithms, as well as, in more detail, the novel algorithm developed at GSOC named Outlier Probability Via Intrinsic Dimension (OPVID) with regards to these constraints. To this extent, we will use both academic and custom benchmarks based on artificial data and historic satellite telemetry to highlight the difficulties as well as provide solutions for choosing the right algorithms and their parameters for the wanted results.

In addition to the analysis of the different algorithms for these benchmarks with mostly predefined features used as the algorithm input, we also want to provide a compact analysis of different features unique to their use case for satellite telemetry as an input to the OPVID algorithm. The results can also be extrapolated for various other algorithms. In an operational use case, these features need to be generic enough to describe every available telemetry parameter and, at the same time, provide a context for the engineers as the automated system should complement the operations team. In the result, we will see that the selection of the features has a large effect on both the false positive and true positive rate and is one of the keys to designing an anomaly detection system for an operational use case.

Nomenclature

ATHMoS Automated Telemetry Health Monitoring System
GSOC German Space Operation Center
LOF Local Outlier Factor (outlier detection algorithm)
LoOP Local Outlier Probability (outlier detection algorithm)
LSTM Long Short Term Memory
ID Intrinsic Dimension (a type of statistical quantity assigned to a data point and data set)
IDOS Intrinsic Dimension Outlier Score (outlier detection algorithm)
OPVID Outlier Probability Via Intrinsic Dimension (outlier detection algorithm)
ROC AUC Receiver Operating Characteristic Area Under Curve

*MCDS (Mission Control and Data handling System Engineer) Mission Operations Engineer, Mission Operations Department, German Space Operations Center, leonard.schlag@dlr.de
†Mission Planning System Engineer, Mission Operations Department, German Space Operations Center
‡Deputy Head of Mission Operations Department, German Space Operations Center

I. Introduction

The problem of detecting outliers in time series data is neither new nor does it lack techniques and algorithms to tackle it. These techniques include many approaches such as using autoencoding neural networks, distance and density based methods, or support vector machines as summarized in [1]. For anomaly detection in satellite telemetry, automated anomaly detection using these approaches is gaining importance [2, 3] as manual inspection of the telemetry is not feasible for thousands of telemetry parameters. Already commonly used static checks using e.g. simple value thresholds are often too coarse to detect subtle behavioural changes.

The constraints for an automated anomaly detection system for satellite telemetry are unique compared to typical use cases only covering a few parameters. The underlying algorithm needs to be generic enough to handle various types of behaviours since e.g. the time series describing a current is vastly different to the one describing a temperature or status of a switch. In addition, there is no labelled or nominal training data available, often necessary as an input for the algorithms, therefore requiring the automatic extraction of training data from an unlabelled set of historic telemetry data. To make the output usable for subsystem engineers overseeing the satellite, it is advantageous to use descriptive features representing comprehensible properties in order to not only flag anomalies, but also give an understanding as to why it was flagged, e.g. the noise being unusually high. Lastly, minimizing the false positive rate of such an anomaly detection system is of great importance and becomes apparent when considering that a modern satellite with 80 000 telemetry parameters would on average trigger more than 200 false alerts a day if the underlying algorithm flagged only 1 false positive per year for each parameter. This would cause the automatic anomaly detection system to be unreliable and create a large amount of overhead as subsystem engineers would have to manually filter out the false alarms from the true anomalies. At the same time, the cost of minimizing the false positive rate should not be too high with respect to the true positive and false negative rate as the anomaly detection system should still be able to outperform static methods with manual inputs.

Considering these constraints, three algorithms are analysed in multiple benchmarks and various feature sets are analysed in a realistic test using historic satellite telemetry. This analysis should outline how anomaly detection algorithms and systems can be chosen, tested, and tweaked considering their usage for satellite telemetry. In the conclusion, first results of ATHMoS, the anomaly detection framework in development as GSOC, are summarized as the core algorithm of ATHMoS, OPVID, is one of the algorithms analysed in this paper.

II. Analysed Algorithms

The following three algorithms will be considered in the remainder of this paper. The input to each is a set of feature vectors and the output are anomaly scores which can be used to flag anomalies.

1) **OPVID (Outlier Probability Via Intrinsic Dimension)** [4]
   OPVID uses a density based evaluation of a quantity known as the *intrinsic dimension* [5, 6] to obtain a probability score similar in meaning to that of the LoOP algorithm. OPVID can also be adjusted using a context set size parameter kc which defines the neighbourhood of points used to calculated the ID score. In addition, a parameter for the reference set size, kr, is introduced to define the size of the local neighbourhood evaluated to derive the outlier probability. The parameter λ scales the error function which is used to obtain an outlier probability within the algorithm. The higher λ, the less sensitive the algorithm becomes.
2) **LoOP (Local Outlier Probabilities)** [7]
   LoOP is a local density based algorithm which is based on the idea of the *local outlier factor* LOF [8] and derives a probability score signifying whether a point is an outlier or not. The context set size parameter equals the reference set size, denoted as k, and can be used to adjust the sensitivity of the algorithm as it specifies how many neighbours are considered for the computation of the local density. Analogue to the OPVID algorithm, the parameter λ is used to scale the error function to obtain an outlier probability.
3) **LOF (Local Outlier Factor)** [8]
   Similar to LoOP, LOF is a local density based algorithm which defines an outlier factor for each data point. Unlike OPVID and LoOP, the LOF-Score does not represent a probability but a factor that can become arbitrarily big. A factor close to 1 describes nominal points within a cluster where as a large factor hints at an anomaly. The parameter k also describes the context and reference set size at the same time and is used to compute the local reachability density defined in [8], thus having a similar meaning as the parameter **k** in the LoOP algorithm.

III. Benchmarks and Numerical Analysis

At first, we want to look at several benchmarks and use their outputs to numerically analyse the LOF, LoOP and OPVID algorithms. The key here is to understand the impact of the variable parameters these algorithms use as their input. With respect to satellite telemetry, the goals we want to achieve by tweaking these parameters and choosing an algorithm are:

- Minimizing the false positive rate (~low scores for nominal data)
- Maximizing the difference in scores between nominal and anominal points for a clear distinction
- Robustness towards change in feature dimensions
- Robustness towards the shape of clusters formed by the feature vectors.

In the following, two different benchmarks are described and used to analyse the three algorithms described in Section III.B with respect to the above goals.

A. Artificial Hyperrectangle Benchmark

The first benchmark we want to look at uses automatically generated random feature vectors as the input. It will be used to highlight the behaviours of the LOF, LoOP and OPVID algorithms with respect to both increasing feature vector dimensionality and an increasingly distinct anominal feature vector. For the benchmarks included in this section, we used $kc = 60,\ kr = 300,\ k = 60$ and $\lambda = 6$, where kc and kr are the context and reference set sizes used for OPVID and k is the set size used for LoOP and LOF.

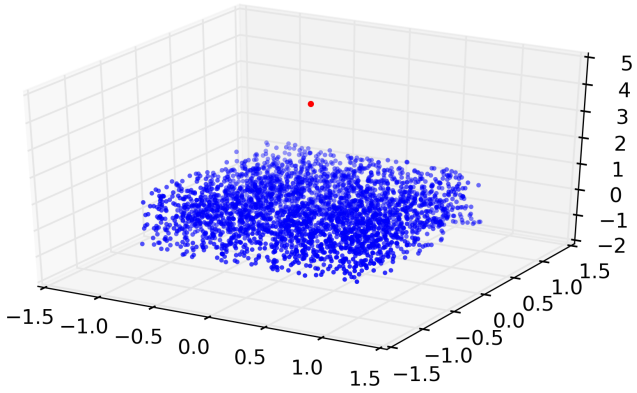


Fig. 1 Example of a 3-dimensional discrete feature vector set describing a cube $\tilde{X}_3 = [-1,\ 1] \times [-1,\ 1] \times [-1,\ 1]$ in blue with an anomaly $\tilde{a}(4.0)$ marked in red. Regarding the analysis of the algorithms, the blue feature vectors in $\tilde{X}_3$ are considered to be nominal features as they belong to the same cluster whereas the red feature vector $\tilde{a}$ represents the anomaly.

The set of feature vectors $\tilde{X}_n$ analysed in this benchmark represents a hyperrectangle in the n dimensions of the feature vector. The third dimension is used to add an anomaly, thus restricting the set of feature vectors to $\tilde{X}_{n\geq 3}$. The hyperrectangle described by this feature vector set is defined as

$$\tilde{X}_n = [a_0,\ b_0) \times [a_1,\ b_1) \times \ldots \times [a_{n-1},\ b_{n-1}),\quad b_i > a_i\ \forall\ i \in [0,\ \ldots, n-1], \tag{1}$$

with its sides in dimension i being limited by an open interval between a_i and b_i.

We discretize $\tilde{X}_n$ using 3000 randomly picked samples from a continuous distribution over the stated intervals. To this discrete feature vector set, we add an anominal feature vector of the same dimensionality

$$\tilde{a}(a) := [\tilde{a}_0,\ \tilde{a}_1,\ \tilde{a}_2, \ldots,\ \tilde{a}_{n-1}] = [0,\ 0,\ a,\ 0, \ldots,\ 0]. \tag{2}$$

As an example, a 4-dimensional anomaly $\tilde{a}(5)$ would write as $\tilde{a}(5) = [0,\ 0,\ 5,\ 0]$.

In the following, $\tilde{X}$ will always denote the discrete feature vector set containing 3000 feature vectors. A 3-dimensional example of this discrete feature vector set and anomaly is visualized in Fig. 1.

1. Dimension Scaling

At first, the influence of the number of dimensions on the algorithms will be analysed. To achieve this, we choose an n-dimensional hypercube with variable n as the nominal feature vector set

$$\tilde{X}_n = [-1,\ 1)^n \tag{3}$$

with the before mentioned discretisation using 3000 feature vectors. The added anomaly (Eq. (2)) is set to 3 times the standard deviation of the nominal feature vector set

$$\tilde{a}(1 + 3 \cdot std(\tilde{X}_n)). \tag{4}$$

To decrease the randomness, the algorithms are run 15 times for each dimension with a randomly generated feature vector set each time and averaged. The input to each algorithm is the union of the nominal feature vector set and the anominal feature vector

$$\tilde{X}_n \cup \tilde{a}(1 + 3 \cdot std(\tilde{X}_n)). \tag{5}$$

The desired behaviour with respect to using an algorithm for satellite telemetry would be a low average score for nominal features in $\tilde{X}_n$ across all dimensions as this would mean it is stable even if features are added or a high dimensional feature space is used. The scalability of the dimension is also of great importance since the extracted feature vectors should be usable for most, if not all, telemetry parameters in a generic manner. As the dimension of the feature vector simulated by increasing dimensions n in Eq. (5) can be interpreted as the complexity and types of behaviours the features can describe, a high dimension ($n \geq 10$) may be required to describe the characteristics of all telemetry parameters combined.

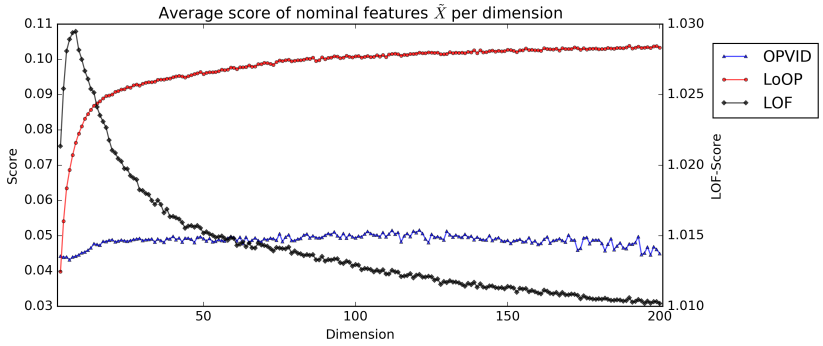


Fig. 2 Average score (15 runs per dimension) of the nominal features calculated for the hypercube feature vector set $\tilde{X}$ described in Eq. (3). The *Score* on the left is the probability based score of OPVID and LoOP while the one on the right hand side is the *LOF-Score*. For the nominal feature vector set, the LoOP score increases with the dimension while the OPVID score is not influenced.

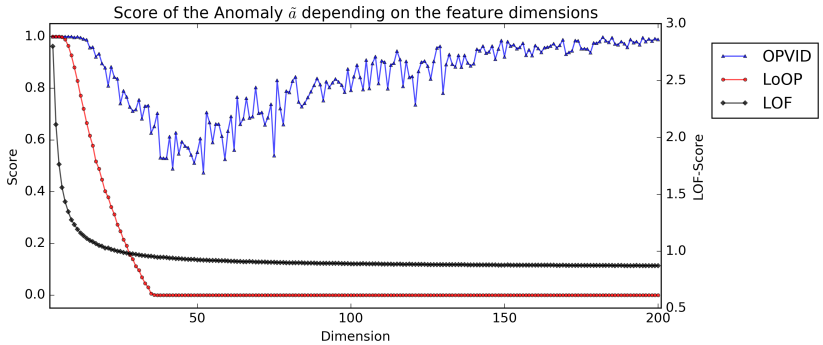


Fig. 3 Average score (15 runs per dimension) assigned to the anominal feature $\tilde{a}$ described in Eq. (4) for increasing dimensions. The *Score* on the left is the probability based score of OPVID and LoOP while the one on the right hand side is the *LOF-Score*. The scores of both LoOP and LOF fall to a low score comparable to that of the nominal data (Fig. 2) while the anomaly probability computed via OPVID remains high even as the dimension is increased to 200.

Analysing the behaviours of the three algorithms under this viewpoint, we see in Fig. 2 that OPVID maintains an almost constant low average score for features contained in the nominal set $\tilde{X}_n$ while LoOP shows a seemingly logarithmic increase in its score with increasing dimensions. While the scores of all algorithms are good for the nominal set, there is a clear discrepancy when looking at the score of the anomaly in Fig. 3. While OPVID is not always over a typical threshold of 90%, the probability based score of the anominal value is high for any dimension and clearly distinguishable from the average score of the nominal features of approximately 0.05 (see Fig. 2). While LOF shows similar characteristics, the probability based score provided by LoOP falls off at higher dimensions and drops to the same order of magnitude as the scores of nominal features at dimensions higher than 40, making it impossible to differentiate between the anominal and nominal feature vectors.

2. Anominal Feature Scaling

To also analyse the behaviour of the algorithms for a more complex shaped cluster and not as distinct anominal feature vector, we want to look at the following 5-dimensional feature vector set:

$$\tilde{X} = [-0.01,\ 0.01) \times [-0.1,\ 0.1) \times [-1,\ 1) \times [-10,\ 10) \times [-100,\ 100). \tag{6}$$

The added anomaly feature vector will increase in its 3^{rd} dimension starting at $\tilde{a}(1)$ and going up to $\tilde{a}(8.5)$. While for $\tilde{a}(1)$ we do not expect the algorithms to flag the feature vector as an anomaly as $\tilde{a}(1)$ extremely close to the nominal set, they should do so with increasing anominaly values. Again, the algorithms are run 15 times for each appended anominal feature vector to reduce randomness.

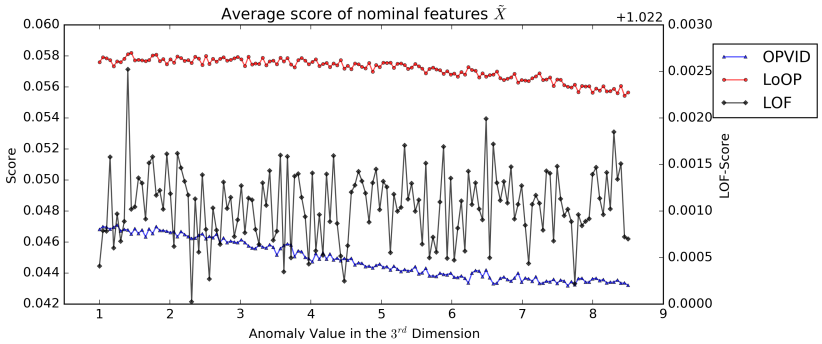


Fig. 4 Average score (15 runs per value of $\tilde{a}$) of the nominal features calculated for the hyperrectangular feature vector set $\tilde{X}$ described in Eq. (6). The *Score* on the left is the probability based score of OPVID and LoOP while the one on the right hand side is the *LOF-Score*. As the value of the appended anomaly $\tilde{a}$ increases, a slight decline is visible since the feature calculation included the anomaly and the larger its value, the less likely it is to be included in a local neighbourhoods used by the algorithms. Overall, all three algorithms show low scores for the nominal feature vector set.

In Fig. 4, OPVID shows the lowest average score for the nominal values compared to LoOP, while LOF seemingly varies a lot. The latter, however, is an effect of the scaling of the LOF-Score axis. While all algorithms show low overall scores for the nominal feature vectors in $\tilde{X}$ with a slight advantage of OPVID over LoOP, the score assigned to the anominal feature vector $\tilde{a}$ is of greater interest It highlights the difference between the algorithms and advantages of OPVID (Fig. 5).

The discrete hyperrectangle described in Eq. (6), on average, had a standard deviation of $\sigma \approx 0.58$ in the 3^{rd} dimension. We would expect to cross the typical threshold of 90% between 3σ and 6σ added on top of the interval border of $\tilde{X}$. This would translate to an anomaly value in the 3^{rd} dimension between 2.74 and 4.48. LoOP reaches the 90% threshold only for a large anominal value of around 9σ due to the highly discrepant interval sizes across the dimensions of $\tilde{X}$ in Eq.(6). The LOF-Score can similarly be interpreted. OPVID shows the most distinct differentiation here and reaches a 90% threshold between 3σ and 4σ.

With respect to the applicability of the algorithms for satellite telemetry, the goal is a low average score for nominal features (nominal telemetry) and a relatively large score for obvious outliers. This goal is best achieved by OPVID within this benchmark.

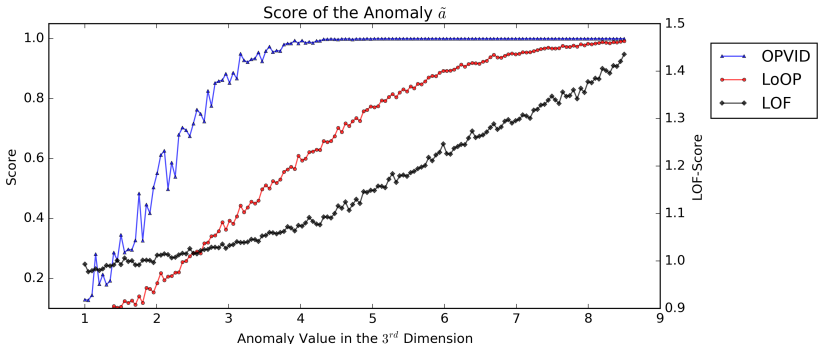


Fig. 5 Average score (15 runs per value of $\tilde{a}$) assigned to the anominal feature $\tilde{a}$ calculated for the hyperrectangular feature vector set $\tilde{X}$ described in Eq. (6). The *Score* on the left is the probability based score of OPVID and LoOP while the one on the right hand side is the *LOF-Score*. $\tilde{a}(1)$ is not distinguishable in all algorithms as should be the case. OPVID is the most sensitive towards the anomaly while both LoOP and LOF only result in high anomaly scores for high values of the anominal feature.

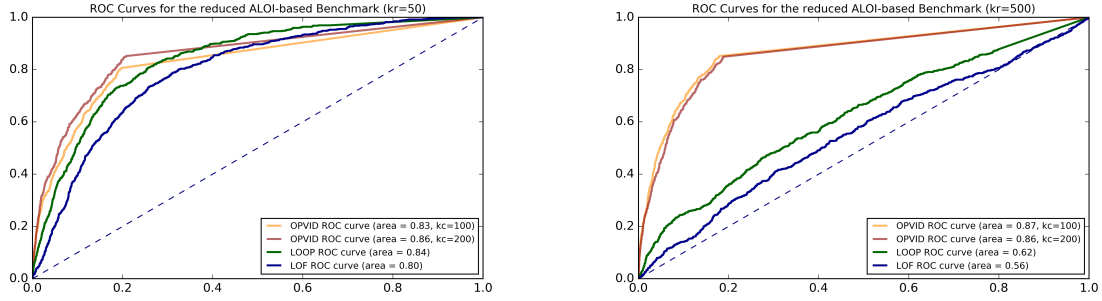
B. ALOI Benchmark

The next benchmark we want to look at is based on ALOI [9], the Amsterdam Library of Object Images. This benchmark is based on features derived from various histograms of object images and was already used to measure the quality of the OPVID [4] algorithm or the IDOS [5] measure on which OPVID is based. This feature vector set has the advantage of having a large amount of labelled feature vectors which are not easy, if not impossible, to classify correctly. For this specific benchmark, we will look at the same subset of 10 000 features with 64 dimensions each, and in total containing 5% anominal feature vectors.

It has already been shown [4] that OPVID shows higher values of the area under the Receiver Operating Characteristic Curve (*Receiver Operating Characteristic Area Under Curve*, ROC AUC), meaning better distinguishability between nominal data and anomalies, for this feature vector set compared to the other algorithms. Here, we want to focus on the actual ROC curve which plots the true positive percentage (sensitivity) against the false positive percentage (specificity) for each algorithm.

In Fig. 6a and 6b, we can see two things:

- The slope of the ROC curve rises faster for OPVID for both high and low values of kr. This can be interpreted as OPVID having a higher accuracy – a threshold can be found s.t. the ratio between true positives and false positives is better, meaning higher, than for LoOP or LOF.
- OPVID can be tweaked better depending on the use case since, for high or low values of kr, we can find a kc such that OPVID has a larger ROC AUC than LoOP and LOF.



(a) **(b)**

Fig. 6 ROC curve for the ALOI benchmark using a total of 10 000 samples with 500 anominal samples. The true positive rate is plotted on the y-axis while the false positive rate is plotted on the x-axis. For LoOP and LOF, a context set size of $k = kr = 50$ was used in 6a and $k = kr = 500$ in 6b. OPVID shows the best results, meaning highest ROC AUC value, followed by LoOP and then LOF w.r.t. this benchmark.

Overall, the steep incline of the curve at low false positive rates is of most importance when using an algorithm for anomaly detection for satellite telemetry. A steep incline means that we can find a threshold which has a high true positive rate and comes at the cost of a low false positive rate. In a first test phase of ATHMoS using OPVID, we were able to see that the qualitative results from this benchmark also transferred to various data sets based on real satellite telemetry as ATHMoS was able to automatically flag true outliers in the past while maintaining a negligible false positive rate.

In addition to highlighting the differences between LOF, LoOP and OPVID, we will use OPVID representatively to outline how the “best" parameter combination can empirically be found. Using the ALOI benchmark, kr and kc were looped between 0 and 500 for kr and 0 and 300 for kc. The ROC AUC is plotted as a measure of quality in Fig. 7. It becomes apparent that for OPVID, a combination of a high value for kr and low value for kc gives the most promising results, meaning highest ROC AUC scores. These tests can further be refined by using labelled features derived from satellite telemetry. This way, we arrived at values of 2% for kc and 10% for kr w.r.t. the total amount of feature vectors in the training feature vector set. While looking at the AUC ROC makes sense as a preliminary method of finding the right parameters, it is also important to look at the false positive rate solely in relation to the threshold used for flagging outliers which we will not cover for this benchmark here.

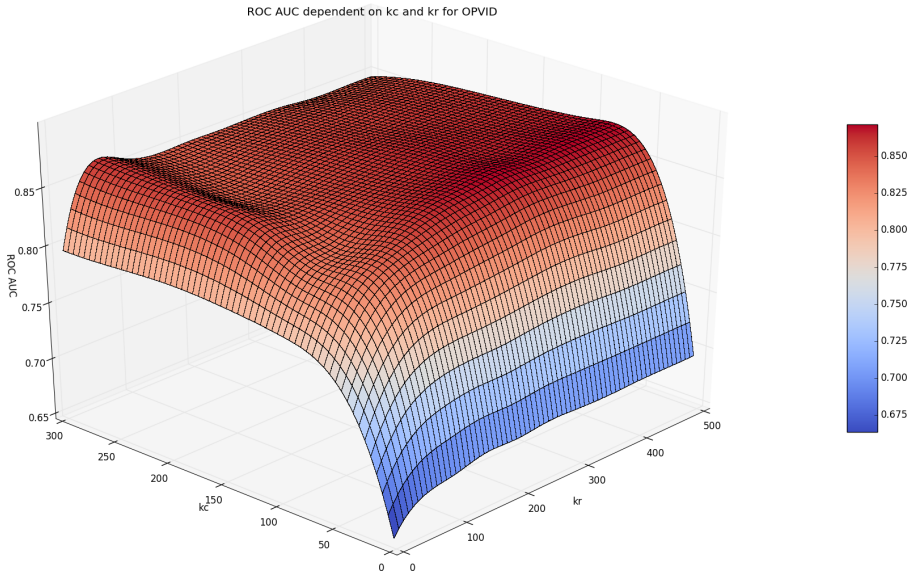


Fig. 7 ROC AUC score of the OPVID [4] algorithm using various combinations of kc and kr. Plots of this type can be used to select the best parameter(s) for the respective anomaly detection algorithms.

IV. Choosing and validating Features for Satellite Telemetry

While the benchmarks in Section III give a general idea of the algorithms performance for various feature sets representing different feature set shapes and dimensions, the next important step towards analysing and using an algorithm in operations is extracting a feature set from time series data describing satellite telemetry. While more novel methods include e.g. orthonormal projections together with dimension reduction techniques as used in e.g. [10] or using a LSTM neural network to train the nominal behaviour of time series data using autoencoding as outlined in [11], other approaches focus on a small feature set of descriptive statistics over a sliding window [12].

We will focus mainly on statistical features and their combinations. The reason for this approach and feature selection is simple: Should an anomaly be flagged, we want to provide subsystem engineers with additional information as to which features of the feature vector triggered the anomaly to provide some context for analysing the telemetry. With automatically trained features generated using e.g. neural networks, the features can not easily be mapped to known quantities (average, standard deviation, ...) and can differ greatly between the parameters. Thus, no information other than the flagged time interval and score could be provided to the engineer as to what went wrong. When using features describing statistics for intervals of the time series data however, the features which contributed most to flagging a time interval as an outlier can be provided, informing subsystem engineers that there was e.g. an unusual amount of noise and

a high amplitude in the anominal time interval.

A. Analysed Feature Sets

Four different sets of features will be used in the following to analyse the impact of different features w.r.t. flagging anomalies and avoiding false positives. Each feature set is calculated for a 90 minute window sliding at steps of 30 minutes, meaning 48 feature vectors are computed for each day in the subsequent analyses. At first, we want to define basic preprocessing of the telemetry which can be applied in Table 1 and basic statistics which can be calculated in Table 2.

Preprocessing	**Description**
Raw	The raw telemetry is not modified except for equidistant linear interpolation of gaps using the smallest time delta > 0 present in the telemetry.
Smoothed	The raw telemetry is smoothed using a Fourier transformation and cutting off high frequencies as described in [4] .
Noise	The *smoothed* telemetry is subtracted from the *raw* telemetry as described above to extract the noise.

Table 1 Preprocessing methods that can be applied to telemetry before extracting statistical features.

Statistic	**Description**	**Statistic**	**Description**
min	Minimum	*75%*	75-percentile
max	Maximum	*midrange*	$\frac{max+min}{2}$
avg	Average	*amp*	Amplitude
std	Standard deviation	*3rd moment*	Third moment
25%	25-percentile	*4th moment*	Fourth moment
50%	Median		

Table 2 Statistics and their abbreviations. Any of the statistics can in our use case be computed for a time series vector describing a 90 minute window of telemetry.

Four feature sets are defined in Table 3 using combinations of preprocessing and statistics. *noise_avg* e.g. stands for the average of the noise.

Basic	*raw_average, raw_max, raw_min, raw_std*
ATHMoS	*raw_std, raw_25%, raw_75%, raw_std, smoothed_avg, smoothed_midrange, smoothed_min, smoothed_max, noise_std, noise_75%*
All	All statistics from Table 2 for each preprocessing method in Table 1. 33 features in total.
Autoencoded + ATHMoS	*ATHMoS* features defined above extended with 10 features extracted using an autoencoding LSTM neural network as described in [11].

Table 3 Four feature sets used representatively for the tests in IV.B.

B. Analysis

The analysis in the following part is done using the OPVID algorithm within ATHMoS as described in [4]. This includes the calculation of the feature vectors for each of the four feature sets defined in Table 3 for 90 minute intervals at a 30 minute sliding window, smoothing, noise extraction, and clustering and cleaning the training data using an

extension of [13]. The feature vectors for the test data are computed in the same manner and the anomaly score is calculated by comparison against the trained models. A standard threshold of 90% (0.9) is used to flag anomalies and visualized in the respective plots (Fig. 9 - 12).

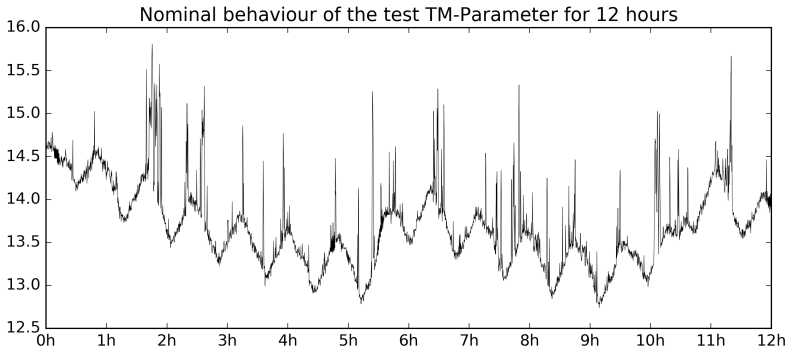


Fig. 8 Example nominal 12 hour time interval of the parameter we used to analyse the different feature sets. It shows a high amount of noise and both short- and long-term periodic behaviour.

An interesting parameter from one of our missions at GSOC was chosen for this analysis and represents the temperature of a reaction wheel. It has both a short- and a long-term periodic behaviour as well as a fairly high amount of noise (see Fig. 8). In early 2015, the parameter has periods of anominal behaviour while being mainly nominal in 2014. The second half of 2014 is used as training data to initialize our models for the different feature sets and the first half of 2015 is tested against this model. While this parameter serves as an example and there are many other kinds of parameter behaviours, the development and initial test phase of ATHMoS at GSOC showed that the results described in this part are similar for a multitude of parameters with different behaviours.

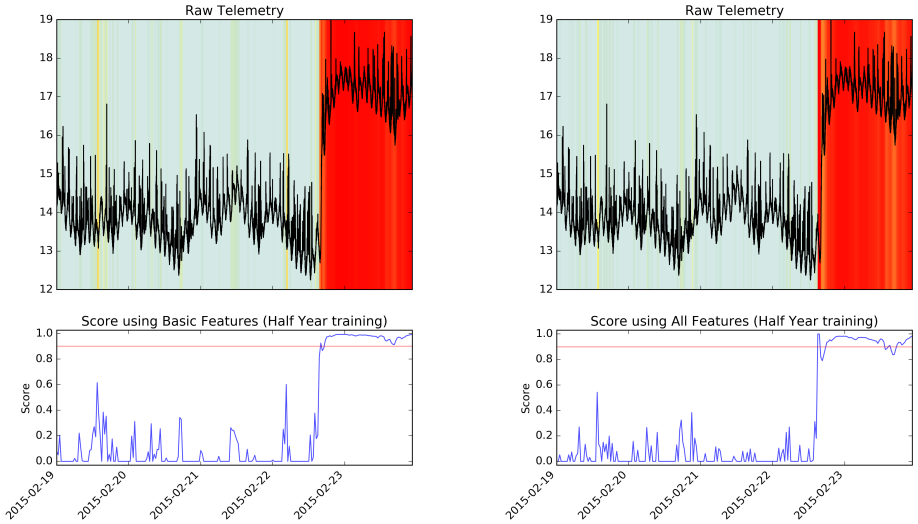


Fig. 9 Anomaly scores computed using OPVID within ATHMoS utilizing the *Basic* features (left) and *All* features (right). The anomaly is clearly visible on 2015-02-22 as the value rises over 16.0 whereas the data leading up to it is nominal. The background was coloured using the anomaly score (plots on the bottom) with green being 0 and red being 1.

In Figures 9 and 10, the resulting anomaly scores in the time before and during the anominal behaviour are shown and compared directly to the anomaly scores computed for this time frame. The effect of the different feature vector sets on the anomaly score can be summarized as follows:

- *Basic* features show some of the largest and longest spikes in the anomaly scores for nominal telemetry leading

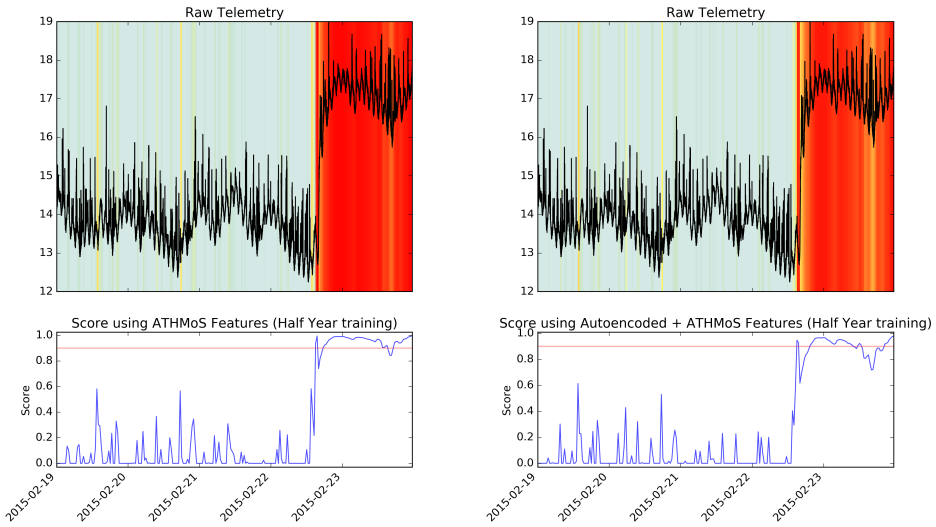


Fig. 10 Anomaly scores computed using OPVID within ATHMoS utilizing the *ATHMoS* features (left) and combined *ATHMoS + autoencoded* features (right). The anomaly is clearly visible on 2015-02-22 as the value rises over 16.0 whereas the data leading up to it is nominal. The background was coloured using the anomaly score (plots on the bottom) with green being 0 and red being 1.

up to the anomaly. This is not desired as it could lead to false positive detections. Figure 11 strengthens this interpretation as the standard deviation, the average, and the maximum values of the anomaly score is higher than for any of the other feature sets in periods where the telemetry is nominal.

- Using *All* features (Fig. 9) dampens the overall scores while still flagging the anomaly. However, there are still many minor fluctuations in the period before the anomaly occurred and the telemetry was nominal. In addition, the anomaly score of the telemetry while it was anominal is almost dampened too much as the score drops below the 90% threshold three times.
- *ATHMoS* features also show fluctuations in the anomaly score during the nominal phase leading up to the anomaly, however not as intense as e.g. the score computed using *Basic* features. The anomaly itself is clearly detected, also returning a high score for the two 90 minute windows before the initial incline happens.
- The addition of the features generated using a neural network in the *ATHMoS + autoencoded* feature set dampens the anomaly score computed using these features heavily, barely surpassing the threshold while the anomaly occurred.

Each of the four feature sets from Table 3 show a good sensitivity towards outliers when used within ATHMoS, while using too little features destabilises the score slightly and too many features might desensitise it too much. This result also reflects the experience gained during the development phase of ATHMoS where the initial use of *Basic* features resulted in a very high false positive rate compared to the currently used *ATHMoS* features.

Of almost greater importance is, as mentioned before, the behaviour of the feature sets and the scores computed using them with regard to nominally behaving telemetry as too many false detections will lead to an anomaly detection system not being trustworthy and usable by those who it was implemented for. A nominal period of one month in June 2015 is considered for this analysis, still using the model trained using the latter half of 2014 to run the data against.

The summary derived from analysing the results for the anominal period transfer well to the nominal test case, depicted in Figure 11 and 12, while certain aspects become more apparent:

- The *Basic* feature set, closely followed by the *All* feature set, show the highest average, standard deviation, and maximum value during the one month of nominal data (Fig. 11). These feature sets can thus be summarized as being overly sensitive, even surpassing the 90% threshold during the nominal period for this parameter.
- The *ATHMoS + autoencoded* features have the lowest average, standard deviation, and maximum value for the month of nominal data.
- *ATHMoS* feature overall show the best sensitivity towards outliers and, for the nominal month of data, a very low average score and standard deviation.

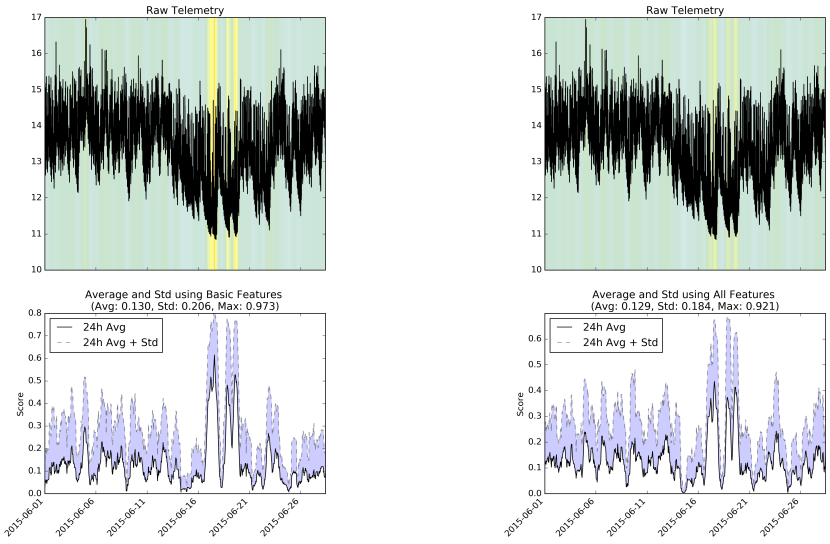


Fig. 11 Averaged (+-12 hours) anomaly scores computed using OPVID within ATHMoS utilizing the *Basic* features (left) and *All* features (right) for 1 month of nominal data. The standard deviation was computed for the same window as the average (+-12 hours) and added on top of the average to visualize how robust the feature sets are for nominal telemetry. The background was coloured using the averaged anomaly score (plots on the bottom) with green being 0 and red being 1. The average, standard deviation, and maximum of the scores for the whole month of data are listed in the plot title.

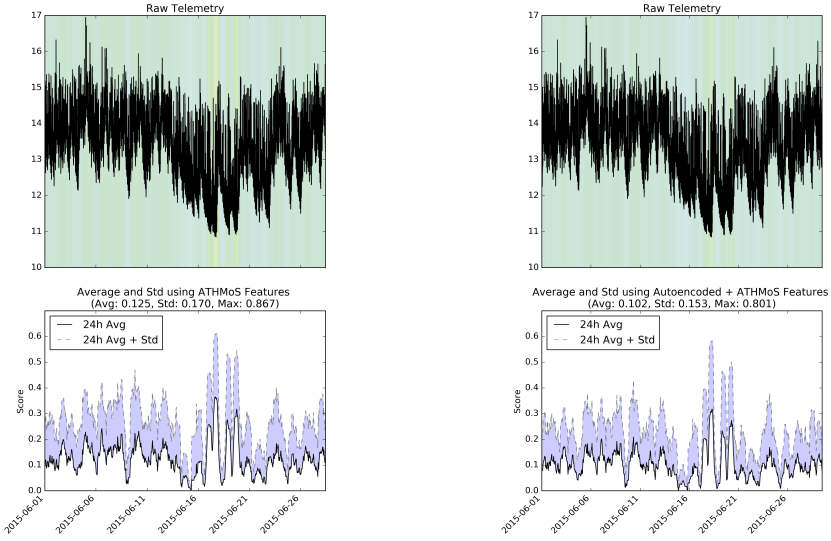


Fig. 12 Averaged (+-12 hours) anomaly scores computed using OPVID within *ATHMoS* utilizing the ATHMoS features (left) and combined *ATHMoS + autoencoded* features (right) for 1 month of nominal data. The standard deviation was computed for the same window as the average (+-12 hours) and added on top of the average to visualize how robust the feature sets are for nominal telemetry. The background was coloured using the averaged anomaly score (plots on the bottom) with green being 0 and red being 1. The average, standard deviation, and maximum of the scores for the whole month of data are listed in the plot title.

V. Conclusion and Application in Operations

The results from Section III show that OPVID has a robust behaviour for low and high dimensional feature sets as well as different distributions along the feature dimensions, thus not requiring heavy post processing of the feature set. At the same time, it allows for a good distinction between nominal and anominal features while keeping the false positive rate low (Sec. III). The *ATHMoS* features in Section IV which were specifically designed for a generic satellite telemetry use case at GSOC have a good sensitivity towards outliers while returning low scores paired with a low standard deviation for nominal time intervals.

As OPVID is the core of ATHMoS, the *Automated Telemetry Health Monitoring System* currently being developed and tested in preparation for operational usage at GSOC, a test using around 800 parameters of an active mission is currently being run in production. This test uses one year of historic telemetry between April 2017 and April 2018 as training data for each parameter and will be extended to multiple missions covering thousands of parameters in the near future. While first results with a naive 90% threshold for flagging outliers caused many parameters to surpass the threshold for time intervals which subsystem engineers would consider as nominal for the respective parameters, this initial implementation was greatly improved by

1) only flagging telemetry if the score of all overlapping windows surpass the threshold and
2) using both a model trained with recent (short timespan) data as well as a model trained with historic (long timespan) data for each parameter and combining the results.

With these two additional measures, true anomalies were still correctly identified while next to no false positives were flagged in the full blown test case as of now. While using ATHMoS in an operational use case will decide how good the performance is, the benchmarks and analyses methods described in this paper were used to find the right algorithm parameters and descriptive features for the very unique use case of satellite telemetry, and are already showing promising results.

References

[1] Pimentel, M.A.F., Clifton, D.A., Clifton, L, Tarassenko, L., “A review of novelty detection", *Signal Processing*, Vol. 99, pp. 215–249, 2014.

[2] “ODVEC Monitoring System Certified and Deployed in JSC Mission Control Center," NASA Press Release, December 2012.

[3] Martínez-Heras, J. A., Donati, A., Kirsch, M. G., and Schmidt, F., “New Telemetry Monitoring Paradigm with Novelty Detection," *Proceedings of the AIAA SpaceOps 2012 Conference*, 2012, pp. 11-15.

[4] O’Meara, C., Schlag, L., Faltenbacher, L., and Wickler, M., “ATHMoS: Automated Telemetry Health Monitoring System at GSOC using Outlier Detection and Supervised Machine Learning", *Proceedings of the AIAA SpaceOps 2016 Conference*, May 2016.

[5] von Brünken, J., Houle, M. E., and Zimek, A., “Intrinsic Dimensional Outlier Detection in High-Dimensional Data", *National Institute of Informatics Technical Report 2015-003E*, Tokyo, Japan, March 2015.

[6] Bouveyron, C., Celeux, G., and Girard, S., “Intrinsic Dimension Estimation by Maximum Likelihood in Isotropic Probabilistic PCA", *Pattern Recognition Letters*, Vol. 32, No. 14, pp. 1706-1713, 2011.

[7] Kriegel, H. P., Kröger, P., Schubert, E., and Zimek, A., “LoOP: Local Outlier Probabilities", *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, November 2009, pp. 1649-1652.

[8] Breunig, M. M., Kriegel, H. P., Ng, R. T., and Sander, J., “LOF: Identifying Density-Based Local Outliers", *ACM Sigmod Record*, Vol. 29, No. 2, pp. 93-104, May 2012.

[9] Geusebroek, J. M., Burghouts, G. J., and Smeulders, A. W., “The Amsterdam Library of Object Images", *International Journal of Computer Vision*, Vol. 61, No. 1, pp. 103-112, 2005.

[10] Barreyre, C., Laurent, B., Loubes, J.-M., Cabon, B., Boussouf, L., “Multiple testing for outlier detection in functional data", *ArXiv:1712.04775*, 2017.

[11] O’Meara, C., Schlag, L., Wickler, M., “Applications of Deep Learning Neural Networks to Satellite Telemetry Monitoring", *Proceedings of the AIAA SpaceOps 2018 Conference*, May 2018.

[12] Verzola, I., Donati, A., Martinez-Heras, J. A., Schubert, M., and Somodi, L., “Project Sibyl: A Novelty Detection System for Human Spaceflight Operations", *Proceedings of the AIAA SpaceOps 2016 Conference*, May 2016.

[13] Ester, M., Kriegel, H. P., Sander, J., and Xu, X., “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", *Kdd*, Volume 96, No. 34, pp. 226-231, August 1996.