Detecting Anomalies in Space using Multivariate Convolutional LSTM with Mixtures of Probabilistic PCA

Shahroz Tariq
Sangyup Lee, Youjin Shin
The State University of New York
(SUNY Korea) Incheon, S. Korea
{shahroz.tariq,sangyup.lee,youjin.shin.1}@stonybrook.edu

Myeong Shin Lee
Okchul Jung, Daewon Chung
Korea Aerospace Research Institute
(KARI) Daejeon, S. Korea
{mslee,ocjung,dwchung}@kari.re.kr

Simon S. Woo*
Department of Applied Data Science
Sungkyunkwan University
Suwon, S. Korea
swoo@g.skku.edu

ABSTRACT

Detecting an anomaly is not only important for many terrestrial applications on Earth but also for space applications. Especially, satellite missions are highly risky because unexpected hardware and software failures can occur due to sudden or unforeseen space environment changes. Anomaly detection and spacecraft health monitoring systems have heavily relied on human expertise to investigate whether they are a true anomaly or not. Also, it is practically infeasible to produce labels on data due to the enormous amount of telemetries generated from a satellite. In this work, we propose a data-driven anomaly detection algorithm for Korea Multi-Purpose Satellite 2 (KOMPSAT-2). We develop a Multivariate Convolution LSTM with Mixtures of Probabilistic Principal Component Analyzers, where our approach uses both neural networks and probabilistic clustering to improve the anomaly detection performance. We evaluated our approach with a total of 22 million telemetry samples collected for 10 months from KOMPSAT-2. We also compare our approach with other state-of-the-art approaches. We show that our proposed approach is 35.8% better in precision, and 18.2% better in F-1 score than the best baseline approach. We plan to deploy our algorithm in the second half of 2019 to actually apply real operation of KOMPSAT-2.

CCS CONCEPTS

• **Computing methodologies → Anomaly detection**.

KEYWORDS

Anomaly detection, LSTMs, Aerospace, Time-series

ACM Reference Format:
Shahroz Tariq, Sangyup Lee, Youjin Shin, Myeong Shin Lee, Okchul Jung, Daewon Chung, and Simon S. Woo. 2019. Detecting Anomalies in Space using Multivariate Convolutional LSTM with Mixtures of Probabilistic PCA . In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19), August 4–8, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3292500.3330776

1 INTRODUCTION

Due to the enormous complexity, and high risk and cost involved with satellite missions, it is crucial to monitor and check the health status of satellite systems during its operation. In particular, an anomaly might be a possible indicator for abnormal operation or behavior of certain parts of subsystems. These anomalies should be detected in order to further prevent the possible cascading failures of a satellite mission, which can possibly jeopardize the satellite mission. However, detecting such anomalies for a satellite is challenging, because an anomaly might be caused not by a single source but multiple sources from a complex interaction of several subsystems. In order to monitor health status and anomalies, the extent of the number of telemetries have to be checked is immense, since different subsystems in a satellite generate more than thousands of telemetries and some of them can be correlated to one another. Moreover, predicting future anomalies with high probability is invaluable to proactively respond to the more serious danger of a satellite and continue on the successful operation of a satellite.

Although anomaly detection has been much deeply researched in other domains, often a rather simple anomaly detection method has been used in a satellite system. While methods such as the Out-Of-Limit (OOL) check methods [7] are simple and easy to implement, it is difficult to detect contextual anomalies with values less than OOL. In addition, anomaly detection in a satellite has required a significant amount of domain knowledge and expertise from operators to define what the nominal and abnormal values and ranges are for each telemetry value (e.g., temperature, power, etc.). Therefore, the traditional approach heavily relied on operators' expertise and knowledge is no longer feasible and scalable. In addition, the lack of labeled data is another challenge, where it is not practical for operators to label thousands of telemetries produced every day. To address these challenges, several approaches [12, 13, 18, 32, 33, 45] have been proposed using various machine learning algorithms to predict satellite anomalies.

ATHMoS [32, 33] developed by German Space Operation Center (GSOC) uses statistical outlier detection method leveraging the historical data to distinguish the possible anomalies. Also, research by NASA-JPL [18] uses nonparametric thresholding method approach with a single-channel LSTM (SC-LSTM) for anomaly detection. They demonstrated the effectiveness by evaluating data from MSL and SMAP. However, their reported contextual anomaly detection performance was not high and it is a single channel model. All these approaches [18, 32, 33] are designed to only be able to predict values for each telemetry individually, requiring individual anomaly detection model per telemetry.

[*] Corresponding Author

ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of a national government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.
KDD '19, August 4–8, 2019, Anchorage, AK, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330776

In this research, we propose a data-driven anomaly detector, based on the novel Multivariate Convolutional LSTM and Mixtures of Probabilistic Principal Component Analyzers (CL-MPPCA) to detect anomalies. Our approach is multi-channel models, which overcomes the limitation from a single channel LSTM model and can detect anomaly more efficiently. In addition, we propose an efficient data sampling method, *Archive Sampling*, which maintains the original characteristics of telemetry data, while providing a significant reduction in input data size to the model.

We evaluated and validated our approach with more than 22 million telemetry data generated from Korea Multi-Purpose Satellite 2 (KOMPSAT-2) satellite by the Korea Aerospace Research Institute (KARI) from May 2013 to February 2014. Further, we compare our approach with other state-of-the-art approaches developed by other space agencies [18, 45] in the world side-by-side and demonstrate the effectiveness of our approach. Our key contributions are summarized as follows:

- We develop a novel data-driven anomaly detector for a satellite system using ConvLSTM and MPPCA (CL-MPPCA).
- We also propose a light-weight Archive Sampling and features, which can reduce the size of a large telemetry dataset to 2% of its originals size, while maintaining high accuracy.
- We used the actual telemetries from KOMPSAT-2 for evaluation and demonstrated the successful performance above 90% precision, recall, and F-1 score.
- We compare our approach with other state-of-the-art anomaly detection approaches by other space agencies (JAXA, NASA, and CNES).
- Our approach is expected to be deployed for actual KOMPSAT-2 operation in the second half of 2019 to improve operational efficiency.

2 BACKGROUND AND RELATED WORK

KOMPSAT-2. Telemetry Data. The Korea Aerospace Research Institute (KARI) [20] is the aeronautics and space agency of South Korea, which has successfully launched several multipurpose satellites including Korea Multi-Purpose Satellite 2 (KOMPSAT-2) [21, 27]. KOMPSAT-2 was launched on July 2006 and the main objectives of KOMPSAT-2 are to provide high-resolution optical national satellite imagery for nation's disaster monitoring, Geographical Information Systems, the composition of maps, agricultural, and oceanographic monitoring. There is a strong need to continue KOMPSAT-2's extended mission, and monitoring the health and detecting anomaly of KOMPSAT-2 is vital. KOMPSAT-2 generates data from more than 3,000 different types of telemetries from various subsystems [27, 31].Therefore, detecting the root causes of anomalies for more than 3,000 different types (high dimension) of telemetries is an extremely challenging issue, as many of them can be correlated. In this work, to process a high dimensional telemetry data, we subdivided the entire telemetries into 2 levels to solve the curse of dimensionality issue. First, the satellite telemetries are divided into subsystem levels, then subsystems are divided into unit levels. Each subsystem has more than one unit and a unit consists of multiple telemetries with continuous or categorical values. The benefit of a 2-level subdivision is that we can better manage the anomaly detector per a subsystem level (multi-channel) than per a telemetry level (single-channel). Our approach is based on a multi-channel model and we will explain more details in a later section.

Anomaly detection in spacecraft systems. Several point and contextual anomaly and outlier detection approaches have been researched [3, 7, 17] for spacecraft systems. Point anomalies are observed as an individual data instance is anomalous with respect to the rest of the data, while contextual anomalies are several anomalous data occur in a specific context. The Out-Of-Limit (OOL) is one of the simplest forms of anomaly detection, where OOL uses unprocessed data and predefined threshold values. There are a number of anomaly detection approaches which provide potential improvements over the OOL methods, such as nearest neighbors methods [4, 5, 23, 26], dimensionality reduction methods [15, 22, 35], clustering based methods [16, 24, 28], expert systems [8, 30, 34, 39], and recently Deep Neural Network (DNN) based methods [18, 46].

Another way to categorize the existing health monitoring and anomaly detection methods are to classify them into 1) knowledge-driven and 2) data-driven approaches. If a comprehensive and accurate knowledge of the domain is available, then knowledge-based methods can be used to diagnose the anomalies in detail. The expert system Intelligent Satellite Control Software DOCtor (ISACS-DOC) is used with Geotail, Hayabusa and Nozomi missions [30]. Numerous other spacecraft also used expert systems for anomaly detection [8, 9, 34, 39]. However, it is rarely the case in modern spacecraft systems to build such a knowledge base from a huge number of sensors in complex systems.

On the other hand, data-driven approaches are built by learning and extracting knowledge from data using various machine learning algorithms. For example, an empirical model can be learned by analyzing and interpreting the past operation data. Then, this model can be used to classify normal or anomalous operation of data in the future. A major advantage of this method is that it is not system dependent and does not require much expert or domain knowledge, as the model will automatically learn and extract knowledge and patterns from data. Because modern spacecraft systems generate a large amount of telemetry data, a data-driven method has become a suitable candidate for anomaly detection methods and it has been actively researched recently by many researchers [10, 11, 14–16, 22, 24, 25, 28, 35, 36, 42–44].

The New Operational SofTwaRe for Automatic Detection of Anomalies based on Machine-learning and Unsupervised feature Selection (NOSTRADAMUS) [12] by Centre National d'Etudes Spatiales (CNES) utilizes the various machine learning methods to detect anomalies for CNES House Keeping Telemetry data. NOSTRADAMUS transforms telemetry data into a vector of features and uses Principal Component Analysis (PCA). Then, anomaly points are classified using a One-Class Support Vector Machine (OC-SVM) trained on nominal data. However, the decision frontier is only described by the support vectors that are part of the learning data points. Hence, the model can be too sensitive generating a large number of false alarms. In this work, we experiment NOSTRADAMUS's OC-SVM with our approach and compare the result. In addition, Yairi et al. [45] used data-driven health monitoring and anomaly detection method based on probabilistic dimensionality reduction and clustering for the Japan Aerospace Exploration

Agency (JAXA)'s small demonstration satellite 4 (SDS-4). This system utilized Mixtures of Probabilistic Principal Component Analyzers (MPPCA) [41] and Categorical distributions (CD). It provides a dependable measurement of anomaly score on novel data to help operators. However, due to the irregularity of sampling periods and the temporal dependency when preprocessing data as well as manual selection of cluster size are left for their future work. We address some of their limitations by developing a better preprocessing method to cope with temporal dependency in data.

Anomaly detection with deep learning. Recently, deep learning has been applied for anomaly detection for spacecraft. German Space Operation Center (GSOC) has developed the modular data framework Automated Telemetry Health Monitoring System (ATHMoS) [32] to efficiently monitor and analyze satellite telemetry. Furthermore, they used an autoencoder to reduce data from input layer into a compressed representation. They show that new feature vectors learned by the autoencoder can detect anomalies on a finer level of detail and reduce the false positive error. However, due to the black-box nature of the autoencoder, it is shown to be challenging to design a sophisticated network.

Hundman et al. [18] at NASA's Jet Propulsion Laboratory (NASA-JPL) proposed the anomaly detection system using Long Short-Term Memory (LSTM) networks with expert-labeled telemetry data from MSL and SMAP spacecraft. A single model is created for each telemetry channel. They also propose an unsupervised and nonparametric anomaly thresholding approach and false positive mitigation strategies. Once the model predicts the value at each step, the prediction error is calculated and then smoothed the sharp spikes in error value that is one of the frequent problems of LSTM based prediction. They use an exponentially-weighted average (EWMA) to generate smoothed errors. The limitation of this paper is that it shows lower performance for detecting contextual anomalies. Also, since it is a single channel model, it would require many models for different telemetries, while our approach focuses on multi-channel (multivariate) model to improve the current limitation.

3 APPROACH

3.1 Preprocessing

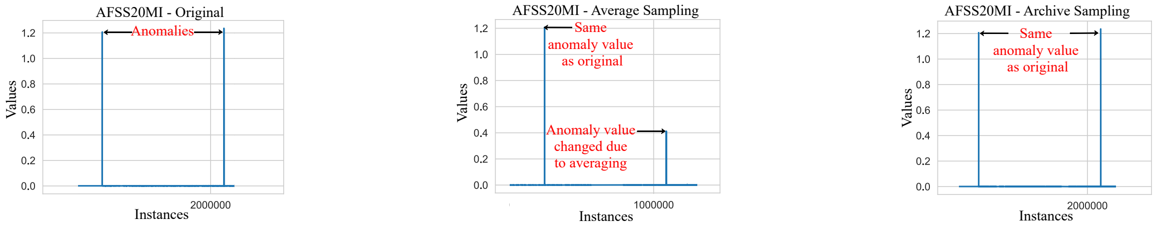
Missing data completion. Due to differing data generation rate from each sensor, certain sensors will produce more telemetries than other sensors and send those to a ground station. Therefore, there will be an unequal number of measured data points for different telemetries given the same time interval. This can cause a large number of missing values in the dataset. Thus, it is critical to decide how to estimate the telemetry values generated at different rates. Should we interpolate and estimate those unmeasured missing values? Or should we sample based on a time window? In order to tackle this issue, we compared different interpolation and sampling approaches to find the best preprocessing method for our CL-MPPCA approach.

Interpolation Methods. There are two different types of telemetries in our dataset. Status telemetries that consist of 1) discrete/categorical and 2) continuous/real values. In order to fill in the missing categorical values, we first copied the telemetry's preceding value. For continuous real values, for example, temperature or voltages, we performed linear interpolation between two consecutive data points to fill out the missing data points. Linear interpolation can preserve the original data points. However, it can produce synthetic interpolated points, which can change the characteristics of underlying anomalous data points. For example, assuming the telemetry value stays between 8 to 12, suddenly due to some anomaly the telemetry value increases to 65535. Then, the missing values between these two data points will be linearly interpolated generating a bunch of new anomalous points in the dataset which are not a true depiction of the real anomaly. Introducing such nonexistent outliers into dataset can significantly mislead the anomaly detection algorithms. Another disadvantage of interpolation is that it is a resource and time-consuming process by filling all missing values. Hence, the size of the dataset increases after linear interpolation. For example, KOMPSAT-2 dataset almost doubled from 1.3GB to 2.59GB after linear interpolation.

Average Sampling. Not only to preserve the raw data values and underlying original data characteristics but also to reduce the size of the dataset, we first applied Average Sampling by averaging out the data within a specific time window into single data value. To find a specific time window to average, we first searched the telemetry that has the lowest sampling rate inside the unit. We calculate the time difference between two consecutive samples of that telemetry and set this as a time window for this unit. Average Sampling can maintain the majority of the characteristic of the original dataset while reducing the file size (from 1.3GB to just 0.18GB in our dataset). However, there is also one major drawback If there are multiple data points for a one-time window and if one of those points is anomalous during the step we take the average. Then, due to averaging out all these points, the real value of anomaly will be changed to something closer to the average. To illustrate this, we provide Fig. 1a to show the two anomalies with AFSS20MI telemetry in the original dataset, while Fig. 1b shows the Average Sampling distorts the original anomaly by averaging. The algorithm of Average Sampling is presented in the Reproducibility Section.

Archive Sampling. To address the problem with Average Sampling, we developed Archive Sampling, which keeps the overall sampled data size small while preserving the original data characteristics. Archive maintains an archive (list) whose size is equal to the number of telemetries in the unit. The data is stored line by line in the dataset file, where each line of the dataset is called a row. In the beginning, the archive is empty, then we read one row from the dataset and populate the archive with the values. Missing values are represented by *NIL*. Then, we read the next row and update the archive with new values. Once there is no missing value or *NIL* in the archive, we will save that row as the first row in our new dataset. Whenever any telemetry value in the archive is updated even with the same previously held value, we save it as a new row in the dataset. By doing so, we eliminated the possibility of adding any new values to the dataset. The algorithm of Archive Sampling is presented in the Reproducibility section.

We were able to maintain the original characteristics of the dataset as shown in Fig. 1c comparing to origin data in Fig. 1a. Also, Archive sampling reduces the data size effectively from 1.3GB to 0.37GB during our evaluation. However, one disadvantage of Archive Sampling is that it might lose the first few rows from the dataset until the archive is fully filled, which is equal to the telemetry with lowest sampling rate in the unit.



(a) Original telemetry with two point anomalies (no sampling/no interpolation)

(b) With Average Sampling, the telemetry value of the second anomaly is decreased due to the averaging effect.

(c) With Archive Sampling, both anomalies and rest of the dataset remain consistent with the original dataset, as no new data points are introduced or removed.

Figure 1: Comparison of different sampling methods using telemetry 'AFSS20MI', where the X-axis indicates data instances in a given time interval and the Y-axis indicates the real values of 'AFSS20MI' telemetry.

3.2 Feature Selection and Computation

Telemetry data are divided into time windows W of 10 minutes based on empirical experiment. We chose a total of 13 different statistical and extrema parameters similar to other research [12, 13, 32, 33] and used different statistical methods to compute the following 13 features: 1) mean, 2) skewness, 3) kurtosis, 4) standard deviation, 5) median, 6) first quartile, 7) third quartile, 8) midrange, 9) median absolute deviation, 10) minimum, 11) maximum, 12) energy (E), and 13) average crossing of W ($\bar{x}_{crossing}$) using $E = \frac{1}{n}\sum_{i=1}^{n} x^2(i)$, and $\bar{x}_{crossing} = \frac{1}{n}\sum_{i=1}^{n} 1_{x(i)>\bar{x}}$, respectively.

Then, the final feature vector $V_{T_i}^n$ for n^{th} time window W for telemetry time series T_i is generated by concatenating different features as shown in Eq. (1) below:

$$V_{T_i}^n = \{mean, \dots, minimum, \dots, energy, \dots\} \quad (1)$$

4 OUR PROPOSED MODEL: CL-MPPCA

The core components of our unsupervised anomaly detection approach use a multivariate Convolutional LSTM (ConvLSTM) [37] combined with a complementary Mixtures of Probabilistic Principal Component Analyzers (MPPCA) [41]. By learning from a large amount of normal telemetry data, the goal of our method (CL-MPPCA) is to effectively predict between normal and abnormal telemetry sequences. By taking inspiration from Hundman et al. [18], we developed an alternative unsupervised thresholding method to automatically assess all different subsystem streams of KOMPSAT-2 and determine whether the prediction errors represent satellite anomalies. The error scores from ConvLSTM and MPPCA are used within a dynamic error thresholding method [18] to acquire the anomaly score. We first explain our ConvLSTM component.

4.1 Predicting with ConvLSTM

Multivariate Model. In our approach, a single model is created for an entire subsystem, where a subsystem contains related telemetries in the ranges of 4 to 35 in our dataset. As observed by Hundman et al. [18], LSTMs struggles to accurately predict n-dimensional outputs with a large n, preventing the input of all telemetry stream into one or a few models [18]. However, Convolutional LSTM (ConvLSTM) has shown to overcome these limitations and thus it is suitable for predicting multivariate time series data as shown by [46].

Since our ConvLSTM model is trained on the entire subsystem, it can provide a prediction for the whole subsystem. However, in order to achieve traceability of anomalies down to channel level, we introduced a separation layer at the end of our model. The separation layer transforms the output of the model into the total number of telemetries inside the subsystem. The telemetries which actually caused the anomaly inside the subsystem will have a different y and $\hat{y}$ values as shown in Fig 3a. Due to this difference, a larger error value is generated which is decisive in detecting anomalies as shown in Fig. 3b. By doing so, the operation engineers will have a holistic view of the system and they can also easily trace down and identify one or more telemetries which caused the anomaly inside the subsystem without putting any extra effort. To avoid overfitting, an early stopping mechanism is applied when validation error starts increasing [6]. A visual representation of our ConvLSTM model is provided in Fig. 4.

Another advantage of our method over a single-channel model is the reduction in the amount of maintenance and computation cost. To provide a non-conflicting environment at least one machine is required for managing a single model. Let's take the AOCS_STA2 subsystem unit inside the KOMPSAT-2 satellite dataset, which is a small-sized subsystem unit, as an example. It has a total of 35 telemetries. The time, effort and money required for maintaining 35 machines each running a single model for one of the 35 telemetries and then regularly retraining them with new nominal data versus maintaining and retraining just a single model for 35 telemetries on a single machine clearly demonstrates the practical advantages of our approach.

Multivariate prediction. The time series T is transformed into a sequence τ containing the historical values of some telemetry i. The length of the sequence l_τ determines how far we want to look into the past for the future predictions. The time series for some telemetry i is $T_i = \{s_1, s_2, s_3, \dots, s_n\}$, where s_j is the j^{th} step in T_i.Also, j^{th} is defined as follows:

$$s_j = \left\{s_j^1, s_j^2, s_j^3, \dots, s_j^f\right\}, \quad (2)$$

where any step s_j in T_i is an f-dimensional vector. Also, f is the size of $V_{T_i}^n$. From Eq. (1) and Eq. (2) $s_j = V_{T_i}^n$, hence:

$$s_j = \{mean, \dots, minimum, \dots, energy, \dots\}$$

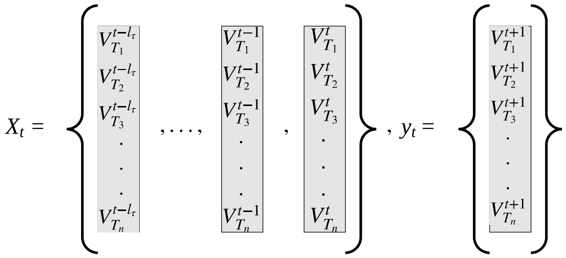


Figure 2: The input and output representation for ConvLSTM

Finally, X_t is a n-dimensional feature vectors of each telemetry in a unit.

$$X_t = \left\{T_1^{\{t-l_\tau,..,t\}}, T_2^{\{t-l_\tau,..,t\}}, T_3^{\{t-l_\tau,...,t\}}, .., T_n^{\{t-l_\tau,..,t\}}\right\}$$

where T_n^t is the feature vector of n^{th} telemetry at timestep t. similarly, y^t is an n-dimensional vector containing the prediction values $(t+1)$ for all the telemetries. The input X and output y of our model are also described in Fig. 2.

$$y_t = \{T_1^{t+1}, T_2^{t+1}, T_3^{t+1}, \dots, T_n^{t+1}\}$$

Training/testing ConvLSTM Model. For the input of ConvLSTM model, we first normalized all the values between 0 and 1. As shown in Fig. 2, sequences of time series data X_t generated by concatenating the rows of telemetry features are provided as an input to train ConvLSTM model. The parameters of the model are trained based on the training dataset X_t and tested with new incoming telemetry values, which are preprocessed in the same way as the training data.

Error calculation. Due to the instability and periodic changes caused by various environmental factors, the telemetry values can vary a lot and it becomes a challenging task to determine if a sequence is nominal or anomalous. As Hundman et al. [18] discussed, Gaussian assumption [1, 38] and distance-based approaches [16, 26] become either problematic when parametric assumptions are violated or involve a high computation cost. We employed a dynamic error thresholding method based on Hundman et al. [18], which solves the aforementioned problem. The error for each prediction ξ_t is calculated using $\hat{y}_t$ and y_t.

$$\xi_t = |\hat{y}_t - y_t| \tag{3}$$

$$\alpha_{score} = \frac{max\,(\xi_\tau) - argmax(\epsilon)}{\mu(\xi_\omega) + \sigma(\xi_\omega)}. \tag{4}$$

A list of historical errors $\xi = [\xi_{t-h}, \dots, \xi_t]$ is calculated by appending ξ_t. ξ is used to assess the current prediction error. The size of ξ is denoted by h. Also, error smoothing is applied to reduce the spikes in error ξ to form $\xi_\omega = [\xi_\omega^{t-h}, \dots, \xi_\omega^t]$, using exponentially weighted moving average method [19]. Due to the lack of label, we opted a threshold based method [18] to acquire an anomaly score (α_{score}) as shown in Eq. (4).

4.2 Predicting with MPPCA

In unsupervised machine learning, dimensionality reduction and mixture models are frequently used to detect anomalies in unlabeled high dimensional data [45]. Therefore, we utilized Mixtures of Probabilistic PCA (MPPCA) [41] to handle high dimensional inputs as well as to complement ConvLSTM.

The output from this model produces an anomaly score between 0 to 1 (1 being the highest probability to be an anomaly). MPPCA is combined with the ConvLSTM model to generate the final output that is presented to the operators for evaluation. To save computational resources, our MPPCA model is made per unit as a multi-channel model. It also makes our MPPCA model consistent with our ConvLSTM model. After calculating anomaly scores from the unit's model, we generate the reconstruction error for each telemetry to obtain the traceability down to the channel level. This reconstruction error indicates which telemetries would be possible anomalies based on predicted anomaly score. We implemented our MPPCA based on Yairi et al.'s algorithm [2].

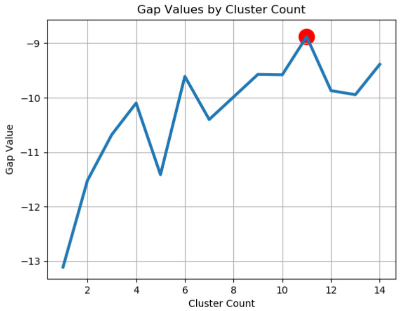
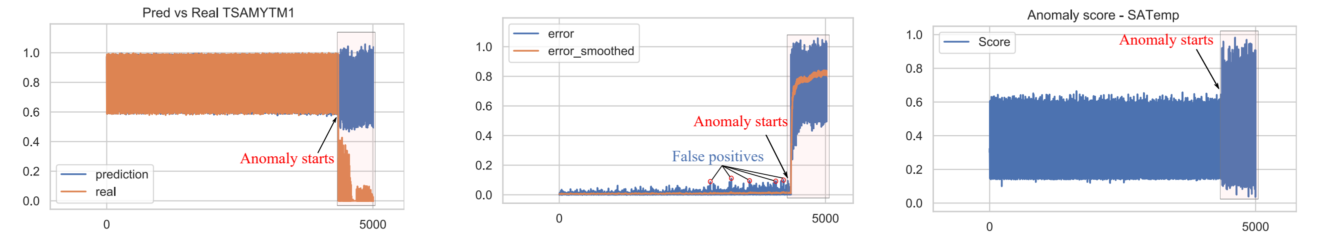


Figure 5: Gap statistics to estimate K, where the X-axis is the number of clusters and the Y-axis is the gap value.

Initializing parameters with K-means clustering. To avoid being stuck in local optima that are poor to the global ones during training, K-means clustering can be used to better initialize the parameters for the MPPCA model before training. However, initializing the number of clusters K is a nontrivial problem as mentioned by Yairi et al. [45]. To estimate the number of clusters, the elbow method, the silhouette method, cross-validation, information criteria, or Bayesian model can be used. Yairi et al. [45] tried some of these approaches but they tended to recommend excessive values of K. Also, the disadvantage of the elbow and average silhouette methods is that they only measure global clustering characteristics. A more sophisticated method is to use the gap statistic [40], which provides a statistical procedure to formalize the elbow/silhouette heuristic in order to estimate the optimal number of clusters. For this reason, we utilized the gap statistics to estimate the K in our dataset. Fig. 5 shows an example of estimating the K with 3-month unit 'SATemp' dataset from May 2013 to July 2013. The estimation of the optimal clusters K is the value that maximizes the gap statistic, where the maximum gap value is at cluster count K=11, as shown in Fig. 5.

Training/testing MPPCA Model. MPPCA is a mixture of probabilistic frameworks, in which all the parameters are trained using maximum-likelihood estimation using an Expectation Maximization (EM) algorithm. After normalization and feature computation in Section 3.2, instead of using sequences of time series data to generate X_t, plain rows of telemetry features are used as an input X_t for the MPPCA model. The parameters of the model are trained with this X_t and are tested with new incoming telemetry values,



(a) ConvLSTM: Ground truth (orange) vs. ConvLSTM's prediction (blue), where the Y-axis is the scaled-down telemetry value of $\hat{y}_t$ (prediction) and y_t (ground truth). When the anomaly starts, there is a sharp drop in y_t; however $\hat{y}_t$ remains consistent resulting in a large error ξ_t.

(b) Error Smoothing: A sharp increase in the smoothed error ξ_ω^t (orange) can be observed as the anomaly starts. The Y-axis represents the error values of ConvLSTM. Normal error ξ^t (blue) tends to result in more false positives (5 FP) as indicated with red circles, while error smoothing produced FP to 0.

(c) MPPCA Anomaly score: The anomaly score increases when contextual anomaly starts and MPPCA accurately detect this contextual anomaly, where the Y-axis is the anomaly score.

Figure 3: A snapshot of three major components (ConvLSTM, Smoothing, and MPPCA) in our anomaly detection system (CL-MPPCA) with telemetry data evaluated in the same time period for 'SATemp' unit in the presence of one contextual anomaly, demonstrating all three components successfully detect the contextual anomaly.

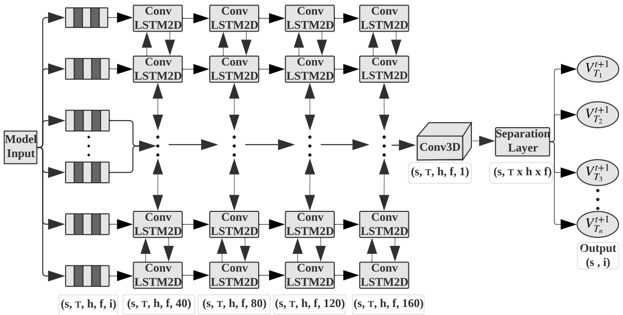


Figure 4: The model of our ConvLSTM, where s, T, h, f and i represents samples, time series, history, number of features and number of telemetries, respectively. (The enlarged figure is also presented in Reproducibility Section.)

which are preprocessed and constructed in the same way as this training dataset.

Anomaly score monitoring and reconstruction error. When new data points are measured and preprocessed, they are passed through to the learned MPPCA model, which generates the log-likelihood value of the observed sample. The anomaly score is defined as minus log-likelihood, where we use this to monitor the anomalies. To be consistent with ConvLSTM method, the anomaly score is normalized between 0 to 1, as shown in the Y-axis of Fig. 3c. In Fig. 3c, we can clearly observe that MPPCA successfully detects the start of the contextual anomaly. To trace down the specific telemetries which affect the highest fraction of anomaly score, we reconstruct $\hat{y}_t$ by calculating the product of the data sample that we want to inspect with the learned weight matrix and finally adding the learned mean of training data. Both the weight matrix and mean of the data are trained by the MPPCA model. With the $\hat{y}_t$ and y_t, the error is calculated using Eq. (3). Then, we inspect the anomalous time period. And we can determine that telemetries with higher error values during that period are more likely to be the cause of the anomaly.

5 EXPERIMENTS AND RESULTS

We conduct extensive experiments to answer the following five important research questions: **(RQ1) Anomaly detection performance .** Can CL-MPPCA outperform over other state-of-the-art anomaly detection methods for spacecraft? **(RQ2) Importance of preprocessing .** Does different preprocessing method matter for anomaly detection performance? Which preprocessing method works the best for CL-MPPCA? **(RQ3) Error smoothing.** Does the error smoothing increase the detection rate and more importantly? does it help in reducing the false positives (FP)? **(RQ4) False positive.** How does the FP-rate effect the performance? **(RQ5) Practicability of feature vectors.** Can we use feature vectors instead of raw or processed data and still achieve similar performance?

5.1 Experimental Setup

KOMPSAT-2 Telemetry Data. We evaluated a total of 22,028,183 telemetry samples collected from 7 subsystem units during KOMPSAT-2's operation from May 2013 to Feb 2014. The detailed statistics of our dataset are provided in Table 1. In the collected dataset, the first 3 months data included more nominal data and the rest of data contained point and contextual anomalies. Therefore, we use the first 3 months telemetry data for training (from May 2013 to July 2013). In this way, we can train our models with normal data values. For testing, we use the rest of longer 7 months data (from Aug. 2013 to Feb. 2014) to evaluate the effectiveness of an algorithm, since we had enough data to evaluate. We applied this training/testing telemetry dataset for all algorithms we compare in this work.

Baseline methods. We compared our CL-MPPCA with other four baseline methods as below:

- **Classification model:** We used One-Class SVM (OC-SVM) [29] and Isolation Forest (IF), which learn the decision boundary from the training set. Also, OC-SVM is adopted by CNES's NOSTRADAMUS [12].
- **Density-based model:** We compare MPPCACD used for JAXA [45]'s SDS-4 to compare our approach.

Table 1: Experimental Dataset, where T indicates telemetry.

Anomaly Type	Sub system	Unit	Anomalous Num. of T (Total Num. of T)	Total Num. of evaluated T Samples
Contextual	TCS	SATemp	1 (4)	3,621,022
	AOCS	CSS	1 (4)	838,481
Point	TCS	UPPTemp	3 (12)	3,433,559
		IPTemp	2 (23)	4,760,176
		CENTemp	1 (11)	1,958,304
	AOCS	STA2	4 (35)	4,096,819
		FSS	4 (6)	3,319,822
Total			15 (96)	22,028,183

- **Prediction model:** We used a single-channel LSTM (SC-LSTM) model developed by NASA-JPL [18].

Machine configurations. We used Intel(R) Xeon(R) CPU E5-2630 0 2.30Ghz with 40GB RAM and NVIDIA GeForce RTX 2070. For implementation, we used Python v3.6, TensorFlow v1.12.0, Keras v2.2.4, Scikit-learn v0.20.2, Visual Studio 2017 for C++, Eigen3 C++ library and MATLAB R2018b.

Evaluation metrics. We use precision, recall, and F1-score for evaluation of all the aforementioned anomaly detection methods. Similar to our CL-MPPCA, MPPCACD also uses multivariate time series data. Hence, we can perform a direct comparison by applying these methods to the units of the subsystem. Whereas, in order to compare the SC-LSTM method with our ConvLSTM in CL-MPPCA, we aggregate the result of SC-LSTM from telemetry level to unit level for a fair comparison.

5.2 Results

(RQ1) Anomaly detection performance. The precision, recall, and F1 score performance of different methods for anomaly detection are reported in Table 2, where the highest performance values are highlighted in bold and the best baseline performance values are indicated by underline. The last "Gain" row in Table 2 reports the improvement (%) of CL-MPPCA over the best baseline method. In Table 2, we observe that (a) our temporal+Density estimation prediction model (CL-MPPCA) achieves higher accuracy of 90.91% F1-score than classification (OC-SVM, IF) and density (MPPCACD) based models, indicating that our dataset has a temporal dependency. As shown in Table 2, SC-LSTM, OC-SVM, and IF show high recall but low precision, indicating most of the predicted labels are incorrect (false positives), when compared to the training labels. (b) Interestingly, MPPCACD performs better than SC-LSTM in F1-score (76.92% vs. 39%). It shows that a large number of FPs in the deep learning model attributes to a low precision score of 25.00%, making its high recall score as less significant. (c) CL-MPPCA performs the best among all the evaluated methods, yielding overall 90.91% F1 score. The gain of our approach over the best baseline (MPPCACD) measured up to 35.8% in Precision. In other words, CL-MPPCA performs much better than baseline methods, as it can model both inter-telemetry correlations and temporal patterns of multivariate time series effectively. A high precision (90.91%) and recall (90.91%) score of CL-MPPCA show that it can detect the most of the anomalies with a very few numbers of false positives and negatives. SC-LSTM yielded the highest recall (91.67%) but low precision (25.00%), which means that it returns many results, but most of its predicted labels are incorrect when compared to the training labels resulting in a low F1-score.

(RQ2) Importance of preprocessing methods. We used the following 3 different preprocessing methods: 1) Linear interpolation, 2) Average Sampling, and 3) Archive Sampling as shown in Fig. 1b. As discussed before, Average Sampling can cause some anomalies to have a very low anomaly score (near nominal value), classifying real anomalies to nominal points. This effectively results in a higher false negative (FN) rate. For example in Fig. 6, the unit 'FSS' has two anomalies. When using Average Sampling as the preprocessing method the second anomaly was given very low anomaly score of 0.01 and our model considered it as a nominal point. However, when the same experiment is performed with Archive Sampling, a very high anomaly score of 0.9871 was computed for the second anomaly as shown in Fig. 6. Therefore, interestingly we find the choice of pre-processing method impacts the anomaly detection performance.

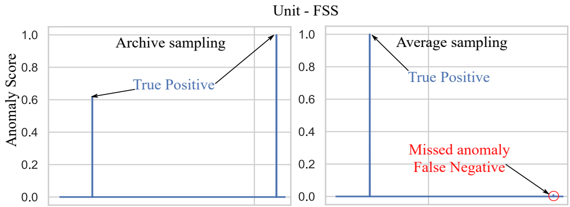


Figure 6: Archive Sampling vs. Average Sampling. The X-axis indicates data instances from 'FSS' unit and the Y-axis is anomaly score from our CL-MPPPCA model. As Archive Sampling preserves the original value, the preprocessed anomaly score shows both point anomalies clearly (Left). However, due to averaging, the second anomaly score with Average Sampling is too low and is not detected (Right).

(RQ3) Error Smoothing. We evaluated the prediction performance of every model with and without error smoothing. After analyzing the results, we concluded that with error smoothing the FP-rate decreased significantly. As shown in Fig. 3b, when we experimented our CL-MPPCA with no error smoothing (blue line) on the 'SATemp' unit, it produced 5 False Positive (FP), 0 false negative (FN) and 1 True positive (TP). However, when error smoothing (orange line) is used, the same model generates $0FP$, $0FN$ and $1TP$.

(RQ4) False positive rate. False positive rate is important because it correlates to the amount of work the operator has to perform in order to identify the true anomalies. As expected, deep learning based models such as SC-LSTM produced more FPs due to high recall and low precision, as shown in Table 2. Whereas density-based methods such as MPPCACD [41, 45] produce less FPs, we can observe that density-based method MPPCACD has a higher precision score (66.67%) than SC-LSTM (25.00%) because of its low FP-rate as shown in Table 2. We believe that the MPPCA module in CL-MPPCA assisted our model to outperform SC-LSTM by having a higher precision (90.91% vs. 25.00%). Our CL-MPPCA outperforms all other baseline methods with an F1-score of 90.91%, which is a gain of over 18% than the baseline method (76.92%).

(RQ5) Practicability of feature vectors. We also quantitatively measured how much storage can be saved with our feature vectors for Archive Sampling ($V_{A}rc.$) while maintaining above 90%

Table 2: Performance comparison of CL-MPPCA with the baseline models.

Method	Precision	Recall	F1 Score
OC-SVM	28.57%	72.73%	41.03%
IF	26.92%	63.64%	37.84%
SC-LSTM	25.00%	**91.67%**	39%
MPPCACD	66.67%	90.91%	76.92%
CL-MPPCA	**90.91%**	90.91%	**90.91%**
Gain	35.80%	- 0.83%	18.2%

Table 3: Dataset size comparison, where d_r, $Arc.$, pli and $V_{Arc.}$ stands for raw data, Archive Sampling, piecewise linear interpolation and feature vector for Archive Sampling respectively. The subsystem unit 'SATemp' is used as an example in this comparison chart. For SC-LSTM and MPPCACD we did not use feature vectors.

Methods	File Size (GB)				Compression	Space
	d_r	$Arc.$	Pli	$V_{Arc.}$	Ratio(C_r)	Saved (ζ)
OC-SVM	1.3	-	-	0.021	61.90%	98.38%
Isolation Forest	1.3	-	-	0.021	61.90%	98.38%
SC-LSTM	1.3	0.37	-	-	3.51%	71.54%
MPPCACD	1.3	-	2.14	-	0.6075%	-64.62%
CL-MPPCA	1.3	-	-	0.021	61.90%	98.38%

F1 score. Since one of the main advantages of using feature vectors is their small footprints compared to a raw or sampled dataset, we used 'SATemp' subsystem unit to experiment different approaches with raw data d_r, Archive Sampling $Arc.$, piecewise linear interpolated Pli and feature vectors $V_{Arc.}$. As we can observe in Table 3, $V_Arc.$ achieves significant saving from 1.3GB to 0.021GB for OC-SVM, IF, and our CL-MPPCA with a compression ratio C_r of 61.90%. Just even using $Arc.$ can provide space saving (ζ) of 71.54%. Since MPPCACD uses Pli, the size increases to 2.14GB. Next, feature vectors further compress sampled data to 0.021GB from 0.37GB. The C_r and ζ are calculated using d_r and compressed data d_c as follows: $C_r = \frac{d_r}{d_c}$, $\zeta = 100 - \frac{d_c}{d_r} \times 100$. Depending on the modeling method, $Arc.$, Pli or $V_{Arc.}$ is used for d_c. We can examine that significant storage gain is obtained from both Archive Sampling and feature vectors while achieving nearly the same anomaly scores and F1 performance. Therefore, we demonstrate that using feature vectors are effective, practical and can be easily adapted for a resource or computation-constrained environments to achieve high accuracy performance.

6 DISCUSSIONS AND LIMITATIONS

Seasonal data can sometimes cause an increase in the FP-rate. This is due to the varying level of temperature and voltage values during a different season and/or time of the year as shown in Fig. 7. The upper and lower graph in Fig. 7 illustrates the original test data from Oct. 2013 to Feb. 2014 in unit 'IPTemp' and their respective anomaly scores from our model. We observed that as the season changes, the temperature sensor values are also affected. This causes FP-rate to increase since resulting anomaly score increases, as shown in Fig. 7. However, seasonal change was not present in our first 3-month training dataset. Hence, we did not take into account this aspect of our current model.

In the future, we plan to gather more data and train our model. Then, we believe it will be better equipped to handle seasonal changes in the dataset. Currently, our model does not inform the human operator about the exact type of anomaly (point or contextual). Also, we did not consider the collective anomaly type. In the future, we plan to implement this capability into our method, so that the operator can better understand the state of the system and detect collective anomalies as well.

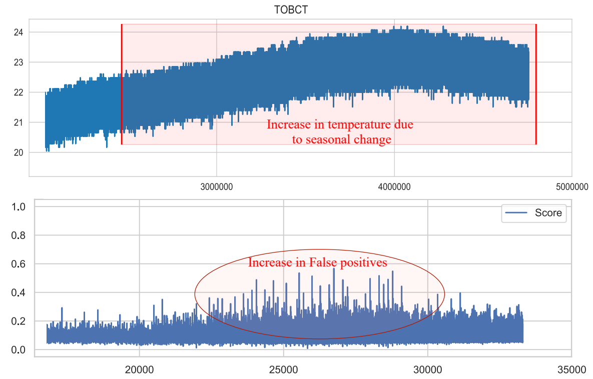


Figure 7: Increase of false positive rates due to seasonal changes, where the X-axis indicates data instance and the Y-axis is temperature (upper graph) and scaled anomaly score (lower graph), respectively. (Note: the X-axis scale slightly is different because it is compressed after Archive Sampling and feature computation.)

7 DEPLOYMENT PLAN

We plan to deploy the presented CL-MPPCA in the second half of 2019 for the actual operation of KOMPSAT-2 to detect anomalies. We will monitor possible anomalies for several hundreds of telemetries of KOMPSAT-2 based on recorded playback data sent from spacecraft. The monitoring period is set to be either every 12 hours or 24 hours to effectively train our anomaly detection system. After validating the effectiveness of our anomaly detection, we plan to extend our approach to other spacecraft such as KOMPSAT-3 and 5. In addition, as a part of an effort to build an integrated ground space operation system, we expect our anomaly detection system will play a major role in monitoring spacecraft health and detect and further predict point and contextual anomaly in advance.

8 CONCLUSION

Manually examining and detecting possible anomalies for spacecraft is a daunting task for humans, as more than thousands of telemetries are generated every day. In this work, we proposed a novel data-driven anomaly detection method based on ConvLSTM and MPPCA and demonstrate the effectiveness of our approach with real KOMPSAT-2 data. Also, our approach outperforms against other state-of-the-art anomaly detection approaches and is planned to be deployed in actual KOMPSAT-2 operation in the second half of 2019. If our paper is accepted, we plan to release our anonymized telemetry dataset used in our research to advance research in this area.

9 ACKNOWLEDGMENTS

We thank anonymous reviewers for their helpful feedback on the draft of this paper. This research was supported in part by Korea Aerospace Research Institute (KARI) grant funded by Ministry of Science and ICT (MSIT) (Satellite Mission Operations), by the MSIT, South Korea, through the ICT CC Program under Grant IITP-2019-2011-1-00783, and by NRF of Korea by the MSIT (NRF-2017R1C1B5076474).

REFERENCES

[1] Subutai Ahmad, Alexander Lavin, Scott Purdy, and Zuha Agha. 2017. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing* 262 (2017), 134–147.

[2] Mathieu Andreux and Michel Blancard. 2016. Mixtures of Probabilistic Principal Component Analysers (MPPCA). https://github.com/michelbl/MPPCA.

[3] Clémentine Barreyre, Béatrice Laurent, Jean-Michel Loubes, Bertrand Cabon, and Loïc Boussouf. 2018. Statistical Methods for Outlier Detection in Space Telemetries. In *2018 SpaceOps Conference*. 2533.

[4] Stephen D Bay and Mark Schwabacher. 2003. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 29–38.

[5] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In *ACM sigmod record*, Vol. 29. ACM, 93–104.

[6] Rich Caruana, Steve Lawrence, and C Lee Giles. 2001. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances in neural information processing systems*. 402–408.

[7] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly Detection: A Survey. *ACM Comput. Surv.* 41, 3, Article 15 (July 2009), 58 pages. https://doi.org/10.1145/1541880.1541882

[8] C Chang. 1992. Satellite diagnostic system: An expert system for intelsat satellite operations. In *Proc. IVth European Aerospace Conference (EAC 91)*.

[9] F Ciceri and L Marradi. 1994. Event diagnosis and recovery in real-time on-board autonomous mission control. In *International Eurospace-Ada-Europe Symposium*. Springer, 288–301.

[10] Dennis DeCoste. 1997. Automated learning and monitoring of limit functions. In *International Symposium on Artificial Intelligence, Robotics, and Automation in Space*.

[11] Dennis DeCoste and Marie B Levine. 2000. Automated event detection in space instruments: a case study using IPEX-2 data and support vector machines. (2000).

[12] Sylvain Fuertes, Gilles Picart, Jean-Yves Tourneret, Lotfi Chaari, André Ferrari, and Cédric Richard. 2016. Improving Spacecraft Health Monitoring with Automatic Anomaly Detection Techniques. In *14th International Conference on Space Operations*. 2430.

[13] Sylvain Fuertes, Barbara Pilastre, and Stéphane D'Escrivan. 2018. Performance assessment of NOSTRADAMUS & other machine learning-based telemetry monitoring systems on a spacecraft anomalies database. In *2018 SpaceOps Conference*. 2559.

[14] Ryohei Fujimaki, Takehisa Yairi, and Kazuo Machida. 2005. An anomaly detection method for spacecraft using relevance vector learning. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 785–790.

[15] Ryohei Fujimaki, Takehisa Yairi, and Kazuo Machida. 2005. An approach to spacecraft anomaly detection problem using kernel feature space. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, 401–410.

[16] Yu Gao, Tianshe Yang, Minqiang Xu, and Nan Xing. 2012. An unsupervised anomaly detection approach for spacecraft based on normal behavior clustering. In *Intelligent Computation Technology and Automation (ICICTA), 2012 Fifth International Conference on*. IEEE, 478–481.

[17] Markus Goldstein and Seiichi Uchida. 2016. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one* 11, 4 (2016), e0152173.

[18] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. 2018. Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding. *arXiv preprint arXiv:1802.04431* (2018).

[19] J Stuart Hunter. 1986. The exponentially weighted moving average. *Journal of quality technology* 18, 4 (1986), 203–210.

[20] Korea Aerospace Research Institute. 1989-2019. KARI website. https://www.kari.re.kr. Accessed: 2019-02-02.

[21] Korea Aerospace Research Institute. 1989-2019. KARI::KOMPSAT satellite website. https://www.kari.re.kr/eng/sub03_02_01.do. Accessed: 2019-02-02.

[22] Minoru Inui, Yoshinobu Kawahara, Kohei Goto, Takehisa Yairi, and Kazuo Machida. 2009. Adaptive limit checking for spacecraft telemetry data using kernel principal component analysis. *Transactions of the Japan Society for Aeronautical and Space Sciences, Space Technology JAPAN* 7, ists26 (2009), Pf_11–Pf_16.

[23] David Iverson. 2008. Data mining applications for space mission operations system health monitoring. In *SpaceOps 2008 Conference*. 3212.

[24] David L Iverson. 2004. Inductive system health monitoring. (2004).

[25] David L Iverson, Rodney Martin, Mark Schwabacher, Lilly Spirkovska, William Taylor, Ryan Mackey, J Patrick Castle, and Vijayakumar Baskaran. 2012. General purpose data-driven monitoring for space operations. *Journal of Aerospace Computing, Information, and Communication* 9, 2 (2012), 26–44.

[26] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. 2009. LoOP: local outlier probabilities. In *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 1649–1652.

[27] Sanguk Lee, Sungki Cho, Byoung-Sun Lee, and Jaehoon Kim. 2005. Design, Implementation, and Validation of KOMPSAT-2 Software Simulator. *ETRI journal* 27, 2 (2005), 140–152.

[28] Ke Li, Yalei Wu, Shimin Song, Yi Sun, Jun Wang, and Yang Li. 2017. A novel method for spacecraft electrical fault detection based on FCM clustering and WPSVM classification with PCA feature extraction. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 231, 1 (2017), 98–108.

[29] Larry M Manevitz and Malik Yousef. 2001. One-class SVMs for document classification. *Journal of machine Learning research* 2, Dec (2001), 139–154.

[30] Naomi Nishigori. 2001. Fully Automatic and Operator-less Anomaly Detecting Ground Support System for Mars Probe'NOZOMI'. In *Proceeding of the 6th International Symposium on Artificial Intelligence and Robotics and Automation in Space (I-SAIRAS), 2001*.

[31] Earth observation resources. 2000-2019. eoPortal website. https://directory.eoportal.org/web/eoportal/satellite-missions/k/kompsat-2. Accessed: 2019-02-02.

[32] Corey OMeara, Leonard Schlag, Luisa Faltenbacher, and Martin Wickler. 2016. ATHMoS: Automated Telemetry Health Monitoring System at GSOC using Outlier Detection and Supervised Machine Learning. In *14th International Conference on Space Operations*. 2347.

[33] Corey OMeara, Leonard Schlag, and Martin Wickler. 2018. Applications of Deep Learning Neural Networks to Satellite Telemetry Monitoring. In *2018 SpaceOps Conference*. 2558.

[34] Mark Rolincik, Michael Lauriente, Harry C Koons, and David Gorney. 1992. An expert system for diagnosing environmentally induced spacecraft anomalies. (1992).

[35] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. 1998. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation* 10, 5 (1998), 1299–1319.

[36] Mark Schwabacher, Nikunj Oza, and Bryan Matthews. 2009. Unsupervised anomaly detection for liquid-fueled rocket propulsion health monitoring. *Journal of Aerospace Computing, Information, and Communication* 6, 7 (2009), 464–482.

[37] X Shi, Z Chen, H Wang, DY Yeung, WK Wong, and WC Woo. [n. d.]. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. arXiv, 2015. *arXiv preprint arXiv:1506.04214* ([n. d.]).

[38] Dominique T Shipmon, Jason M Gurevitch, Paolo M Piselli, and Stephen T Edwards. 2017. Time Series Anomaly Detection; Detection of anomalous drops with limited features and sparse examples in noisy highly periodic data. *arXiv preprint arXiv:1708.03665* (2017).

[39] Donald P Tallo, John Durkin, and Edward J Petrik. 1992. Intelligent fault isolation and diagnosis for communication satellite systems. *Telematics and Informatics* 9, 3-4 (1992), 173–190.

[40] Robert Tibshirani, Guenther Walther, and Trevor Hastie. 2001. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63, 2 (2001), 411–423.

[41] Michael E Tipping and Christopher M Bishop. 1999. Mixtures of probabilistic principal component analyzers. *Neural computation* 11, 2 (1999), 443–482.

[42] Takehisa Yairi, Yoshinobu Kawahara, Ryohei Fujimaki, Yuichi Sato, and Kazuo Machida. 2006. Telemetry-mining: a machine learning approach to anomaly detection and fault diagnosis for space systems. In *Space Mission Challenges for Information Technology, 2006. SMC-IT 2006. Second IEEE International Conference on*. IEEE, 8–pp.

[43] Takehisa Yairi, Minoru Nakatsugawa, Koichi Hori, Shinichi Nakasuka, Kazuo Machida, and Naoki Ishihama. 2004. Adaptive limit checking for spacecraft telemetry data using regression tree learning. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, Vol. 6. IEEE, 5130–5135.

[44] Takehisa Yairi, Takaaki Tagawa, and Noboru Takata. 2012. Telemetry monitoring by dimensionality reduction and learning hidden markov model. In *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space*.

[45] Takehisa Yairi, Naoya Takeishi, Tetsuo Oda, Yuta Nakajima, Naoki Nishimura, and Noboru Takata. 2017. A Data-Driven Health Monitoring Method for Satellite Housekeeping Data Based on Probabilistic Clustering and Dimensionality Reduction. *IEEE Trans. Aerospace Electron. Systems* 53, 3 (2017), 1384–1401.

[46] Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, and Nitesh V. Chawla. 2018. A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data. *CoRR* abs/1811.08055 (2018). arXiv:1811.08055 http://arxiv.org/abs/1811.08055

A REPRODUCIBILITY

This section contains the necessary information for reproducing our method and the baseline methods.

A.1 Algorithms

The algorithms for Average Sampling and Archive Sampling are provided in Algo. 1 and Algo. 2 respectively.

Algorithm 1 Average Sampling

```
$T_{max}$ ← telemetry
$I_{max}$ ← 0
$L_{Tel}$ ← list of telemetries
for $Tel_{cur}$ in $L_{Tel}$ do
    $I_{cur}$ ← $\Delta UTC$ in $Tel_{cur}$
    if $I_{cur} > I_{max}$ then
        $Tel_{max}$ ← $Tel_{cur}$
        $I_{max}$ ← $I_{cur}$
    end if
end for
$L_{buf}$ ← array, telemetry values in the unit
$L_{num}$ ← array, number of values added for each telemetry
for each row in $RawData$ do
    $T_{row}$ ← measured telemetry values in this row
    if $Tel_{max}$ has no measured data in this row then
        if there's already $T_{row}$ in $L_{buf}$ then
            Update $L_{buf}$ with adding $T_{row}$ with previous values
            Update $L_{num}$ with number of values added
        end if
    else
        $L_{avg}$ ← average for each telemetry values in $L_{buf}$
        Save $L_{avg}$ for Average Sampling
    end if
end for
```

A.2 Detailed hardware and software Configuration.

In order to run deep learning models, we used TensorFlow v.1.12.0 with Keras v2.2.4 on NVIDIA GeForce RTX 2070. NVIDIA GeForce RTX 2070 does not support CUDA v9.0 and the python package for TensorFlow v.1.12.0 is only available for CUDA v9.0. So, we complied TensorFlow from the source for v.1.12.0. with CUDA v.10, cuDNN v7.4.2.24 and python 3.6 on windows 10 [1].

Due to the large file size of the datasets, we always read files in chunks to reduce memory (RAM) usage. For the calculation of different statistics for feature vector, we used DataFrame from pandas v.0.24.1 library, which provides some the basic methods such as mean, max, min, skewness, etc. We implemented the statistics which were not available inside the library such as Energy or average crossing, etc.

We downloaded the code for SC-LSTM and MPPCACD for comparison from the following [2] and [3] links respectively.

Table 4: Configuration for ConvLSTM

Layer Name	Units	Filters	Kernel size	Padding	Return sequences	Activation
Conv LSTM2D	-	40	(3,3)	Same	True	-
Batch Normalization	-	-	-	-	-	-
Conv LSTM2D	-	80	(3,3)	Same	True	-
Conv LSTM2D	-	120	(3,3)	Same	True	-
Conv LSTM2D	-	160	(3,3)	Same	True	-
Conv3D	-	1	(3,3,3)	Same	-	-
Flatten	-	-	-	-	-	-
Dense	no. of Telemetries	-	-	-	-	Linear

Scikit-learn library v.0.20 is used for the implementation of Principal Component Analysis (PCA), OneClass-SVM (OC-SVM) and Isolation Forest (IF). Parameters used for PCA are: n_component=0.95, rest are set as default. Parameters used for OC-SVM are: (nu=0.1, kernel='rbf', gamma='auto'), rest are set as default. Parameters used for IF are: IsolationForest (behaviour='new', max_samples=100, random_state=32, contamination='auto'), rest are set as default.

The configurations for ConvLSTM are given in Table. 4. A detailed visual representation of ConvLSTM models is given in Fig. 8. We used Keras library to build this model.

Algorithm 2 Archive Sampling

```
$L_{arc}$ ← Archive(list) for each telemetry in the unit
for each row in $RawData$ do
    $T_{row}$ ← measured telemetry values in this row
    if $L_{arc}$ is not full then
        Update $L_{arc}$ with new values $T_{row}$
        if Updated $L_{arc}$ is full then
            Save updated $L_{arc}$ as the first row of the new dataset
        end if
    else
        if $T_{row}$ are different from the values in $L_{arc}$ then
            Update $L_{arc}$ with new values $T_{row}$
            Save updated $L_{arc}$ to the new dataset
        end if
    end if
end for
```

A.3 Links

[1] https://medium.com/@amsokol.com/update-2-how-to-build-and-install-tensorflow-gpu-cpu-for-windows-from-source-code-using-bazel-61c26553f7e8

[2] https://github.com/khundman/telemanom

[3] https://bitbucket.org/thetak11/dimred-cluster

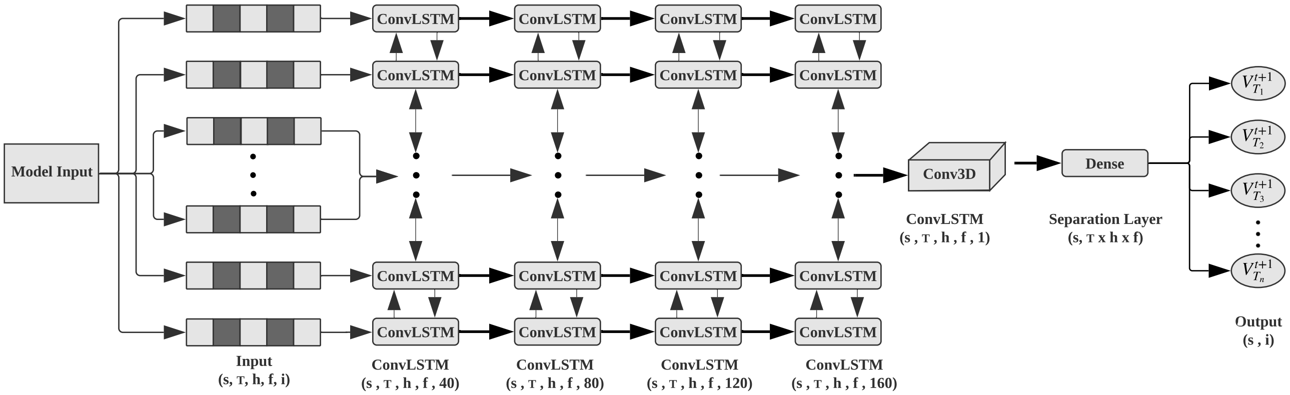


Figure 8: Enlarged figure of Fig. 4.