SOM-Based Anomaly Detection & Localization for Space Subsystems

Maia Rosengarten, Dr. Sowmya Ramachandran
Stottler Henke Associates, Inc., San Mateo, CA USA
mrosengarten@stottlerhenke.com

Abstract. The aim of this paper is to contribute to machine-learning technology that expands real-time and offline Integrated System Health Management capabilities for future deep-space exploration efforts. To this end, we have developed Anomaly Detection via Topological feature-Map (ADTM), which leverages a Self-Organizing Map (SOM)-based architecture to produce high-resolution clusters of nominal system behavior. What distinguishes ADTM from more common clustering techniques (e.g. k-means) is that it maps high-dimensional input vectors to a 2D grid while preserving the topology of the original dataset. The result is a 'semantic map' that serves as a powerful tool for uncovering latent relationships between features of the incoming data points. We successfully modeled and analyzed datasets from a NASA Ames Research Center Graywater Recycling System which documents a real hardware system fault. Our results show that ADTM effectively detects both known and unknown anomalies and identifies the correlated measurands from models trained using just nominal data.

Keywords: Self-Organizing Map, Anomaly Detection & Localization, Integrated System Health Management

1 Introduction

Integrated System Health Management (ISHM) technologies are mission-critical for space exploration. Space habitats are made up of a complex web of subsystems, and the rising demand for rapid fault detection and response in deep-space habitats calls for autonomous monitoring software that can run on board. In particular, communication delays between onboard crews and Earthbound experts (lasting up to 44 minutes) could make the difference between a successful and failed mission, risking the loss of both equipment and crew [1]. Expansion of both machine learning and data mining techniques in this field is therefore of the utmost importance to ensuring mission safety.

In this paper we discuss the application of a semi-supervised approach to anomaly detection and localization called Anomaly Detection via Topological feature-Map (ADTM), which combines a Self-Organizing Map (SOM) for anomaly detection with a Random Forest of Decision Trees to identify the most salient measurands contributing to data flagged as anomalous. Our research has largely been inspired by a successful body of work leveraging SOMs for anomaly detection within the aeronautics domain [2]. Our contribution is the application of this approach to the space domain.

To the best of our knowledge, the use of SOMs in conjunction with decision trees for system health monitoring has never been applied to the space domain before.

The remainder of this paper is organized as follows. *Section 2* reviews related research employing machine learning and statistical techniques for anomaly detection. *Section 3* provides the background for Self-Organizing Maps. *Section 4* provides the technical details and methods of our ADTM model. *Section 5* describes the experiment we ran on NASA ARC subsystem data to test the feasibility of ADTM within the space ISHM domain. *Section 6* concludes our work with a summary of key research findings and plans for future work.

2 Related Work

The focus of this work was on unsupervised anomaly detection for discrete sequences of subsystem data using SOM-based models trained on nominal subsystem behavior. Similar approaches to anomaly detection have been applied in existing research. Principal Component Analysis has been a widely used algorithm for anomaly detection across a wide breadth of applications, including diagnosing offshore wind turbines [3], cyber networks [4], and space telemetry [5]. Furthermore, Gaddam used a supervised approach to anomaly detection by combining K-Means clustering with ID3 decision tree classification [6]. The classification decisions across the clusters and decision trees were combined for a final decision on class membership. The main drawback of such an approach in the space domain is the limited availability of labeled fault data needed for training and validation.

NASA Ames Research Center (ARC) uses k-means and density-based clustering techniques for system monitoring in its IMS and ODVEC software systems [7]. Similarly, Gao, Yang, and Xing used a K-Nearest-Neighbor (kNN) approach for anomaly detection of an in-orbit satellite using telemetry data [8]. SOMs have been used for fault detection and diagnosis in several industries. Datta, Mabroidis and Hosek combine SOMs with Quality Thresholding (QT) to refine the resolution of clusters learned by SOMs within the semi-conductor industry [9]. Similarly, Tian, Azarian, and Pecht train a SOM on nominal cooling fan bearing data but use a kNN approach in place of the Minimum Quantization Error (MQE) to assign test data anomaly scores based on their distance to centroids learned by the kNN model [10]. Cottrell and Gaubert apply anomaly scores to aircraft engine test data using the MQE approach that we have used in this paper (see *Section 4*) and leverage the visualization capabilities of SOMs to visualize the transition states of engines from run-to-failure datasets [2].

ADTM contributes to this existing bed of clustering research by combining a Self-Organizing Map in combination with Extra Tree Classifier for both detecting and localizing faults, which has rarely (if at all) been used in the ISHM space domain.

3 Self-Organizing Map Background

Also known as a Kohonen map, a Self-Organizing Map (SOM) is a two-layer artificial neural network (ANN) that uses unsupervised learning to produce a low-dimensional representation of the training samples [11]. The goal is to transform incoming inputs to a 1- or 2-dimensional map in a topically ordered fashion such that points that are close together in the higher-dimensional input space are also close together in the lower-dimensional output space. This mapping allows us to detect patterns of normal or anomalous behavior in a system, as different types of behavior map to different output units, called "neurons."

Specifically, the N-dimensional input data is fed into the SOM in the first layer and fully connected to a lattice of $(l\ x\ p)$ output neurons O_i in the second layer [10]. Each neuron O_i is associated with a N-dimensional weight vector w_i. We represent O_i by a two-dimensional coordinate of its position in the $(l\ x\ p)$ grid, e.g., $O_i = (x_i, y_i)$. The values of l and p are parameters that are tuned during model validation. Based on the literature [10], we chose l=p= $\sqrt{}(5\sqrt{N})$, though we intend to further tune these parameters in future work. Unlike k-means, the clusters learned during SOM training are topologically ordered through a competitive learning rule.

The topological ordering happens with the following training process: each input vector $m \in M$ is compared with the weight vector w_i associated with each neuron O_i. The closest O_c is chosen as the winner, or 'Best Matching Unit' (BMU), where 'close' is defined by a distance function between the input vector m and the closest w_c associated with O_c. The smallest distance is called the *Minimum Quantization Error* (MQE). Each BMU in the output layer is related to an entire *neighborhood* of neurons through a 'neighborhood function' h(c,k) that computes the relation between the BMU O_c and neuron O_k. The weight vectors within a neighborhood are updated in proportion to their distance to the BMU in the 2D output lattice. Because entire neighborhoods of related neurons get updated in the direction of the input data that is closest to them, the topology of the N-dimensional input space is preserved in the 2-dimensional output space.

Our research used open-source Python libraries for data processing and building a baseline SOM model. Though there are several SOM-based open source libraries available, we chose Somoclu [12] because it leverages a highly parallel implementation in the *C* programming language. Without performing cross-validation for hyperparameter tuning, our SOM-based anomaly detectors still showed promising results in the experiment detailed in *Section 5*. This suggests a significant opportunity for additional performance and efficiency gains through fine-tuning our baseline algorithms in future work.

4 Methods

At a high level, our methods use a SOM trained on nominal subsystem behavior to identify anomalous data, followed by a Random Forrest to identify the salient measurands implicated in the flagged anomaly. ADTM is implemented with the following procedure (each step is detailed in the sections that follow):

1. Given nominal data and fault data sets for testing, divide the nominal data into a training and a test set.
2. Normalize all the data with decimal scaling, using the training data as the scaling reference.
3. Train one SOM per subsystem using nominal training data.
4. For all data sets (training sets, nominal test sets, fault test sets), calculate the MQE for each sample point.
5. Set the k^{th} and (1- k^{th}) percentiles of the MQE scores of the training data as the nominal MQE thresholds for flagging anomalies, where $0 < k < 1$. In our experiments k=0.99.
6. For each data point in each test set, flag the point as anomalous if its MQE is $<$ lower nominal MQE threshold or $>$ the upper nominal MQE threshold.
7. Find the most salient measurands contributing to data flagged as anomalous with a supervised feature extractor, ordered by importance.

Steps 1 – 2 are detailed in *Section 4.1*. Step 3 uses the SOM training process described in *Section 3.* Steps 4 – 6 are detailed in *Section 4.2,* while Step 7 is detailed in *Section 4.3.*

4.1 Data Processing

Because the SOMs require numerical data, we converted categorical variables (e.g., “ON/OFF”) to quantitative variables (e.g., 1/0). We also dropped the columns related to the timestamp of data collection, as we were not concerned with multi-scale time-series analysis. Such analysis is a research focus of future work, however, specifically for the purpose of conducting cross-subsystem analysis given datasets measured in different timescales. Additionally, we normalized the data to prevent measurands with larger ranges from out-weighing measurands with smaller ranges. We used decimal scaling to scale the values so that all values fell within the range -1 to 1 [13]. Normalization was done on the training sets for each subsystem, and the test sets were scaled relative to this normalization.

4.2 Anomaly Detection via MQE

Once trained on nominal data, the SOM maps new data seen during testing to the most similar weight vector w_c of the output neurons O_i, using Euclidean Distance as the similarity metric. A low MQE implies that the new sample closely aligns with a previously seen sample from the training data and is therefore nominal, whereas a higher MQE connotes that the point is anomalous, either because it contains a true fault or because it captures novel nominal behavior unseen during training. We defined a range of nominal MQE scores and classified all samples as anomalous during testing if they fell outside that range. The range was chosen by re-running the training data through an already-trained SOM and setting the 1-percentile value and the 99-percentile value of the resulting MQEs as the lower and upper bounds respectively. Admittedly, these thresholds were chosen rather arbitrarily from our observations of the available data. We intend to include a more principled approach to threshold tuning in future work and anticipate doing so will improve the generality of our results.

4.3 Anomaly Localization via Supervised Feature Extraction

In addition to identifying regions of anomaly, it would be helpful to localize the anomalies to a small subset of the measurands that explain observed behavior deviation and best distinguish between the two regions (anomalous vs. non-anomalous). For this, we rely on the insight that the two regions can be treated as two classes and supervised classification methods can be used to identify the features that distinguish them. For this analysis, we are not concerned about the accuracy of anomaly identification, i.e., the external consistency of anomaly detection with respect to ground truth. All we are concerned with is determining the features that accurately separate two given segments of data (i.e., internal consistency). The data points labeled as anomalies are grouped into one class, and the weight-vectors learned by the SOMs during training form the data for the second class. This highlights one of the benefits of the SOM-based approach: the weight vectors of the SOM are effectively a reduced representation of the training data and lead to efficient storage for future analysis.

A number of supervised feature extraction approaches are available, though we chose Extra Tree Classifier [14], which is a variant of the Random Forest approach [15]. Our experiment in *Section 5* demonstrates the utility of this approach to anomaly localization, though we intend to experiment with other techniques such as Recursive Feature Elimination (RFE) [16] in future work. Our anomaly localization goal was to identify subsets of measurands that contributed the most to deviation in anomalous data. For this we employed a Random Forest (RF) to classify nominal and anomalous data points and output the measurands that resulted in the greatest reduction in Gini Impurity scores across all decision trees employed [15]. We used the default parameters that came from a Python machine-learning package, in which the number of trees was set to 10. We intend to tune this parameter (e.g., employing 100s of trees) to further improve our anomaly localization capabilities in future work.

The weights associated with the SOM output neurons are known as "codebook vectors," as they represent prototypical nominal activity learned from the training data. Thus, we labeled these codebook vectors as "nominal" for our Extra Tree Classifier model. Similarly, we labeled the data samples from our test sets that were flagged as anomalous as the class "anomaly." Finally, we trained a RF on the labeled data and output the list of measurands that resulted in the best splits between the two classes, ranked by their (normalized) reductions in GI scores across all trees. For this paper, we arbitrarily chose the subset of measurands with a feature importance score of at least 10 to characterize the anomalies from each test set, though this threshold is an additional hyperparameter that we will tune in future work.

5 Experiments and Discussion

We divide the results of our experiment into the following three subsections: *Data Collection, Anomaly Detection Analysis, Anomaly Localization Analysis.*

5.1 Data Collection

We acquired data of a Graywater Recycling System from NASA Ames Research Center installed at Stanford University. This data documented a real system failure that propagated across two interconnected subsystems. The Forward Osmosis (FO) membrane became fouled with a bacterial sludge, and the system shut down due to a low OA tank float alarm. The data we received decomposed the Graywater Recycling System into two subsystems, "Subsystem 1" and "Subsystem 2." For each subsystem, we received two days' worth of nominal data and four days of faulty data. We divided the nominal data into a training set and a nominal test set for each subsystem, the latter of which was used to compare against the fault test sets. The shapes of the datasets used for training and testing are described in Table 1. Although both Subsystem 1 and Subsystem 2 were running for the same length of time during the October experiments, Subsystem 1 has significantly more data points than Subsystem 2 due to differences in the sampling rate for each subsystem.

Table 1. Datasets used for training and testing SOMs for Subsystems 1 & 2

Subsystem	Train data (#rows, # features)	Test data (#rows, # features)
Subsys1 SOM	(47031, 32)	fault: (274100,32) nominal: (23643, 32)
Subsys2 SOM	(789, 7)	fault: (4595, 7) nominal: (394, 7)

Beyond detecting the fault, our algorithm was also able to output the specific sensors that contributed most to the anomaly, as shown in *Subsection 5.3*.

5.2 Anomaly Detection Analysis

For each test set, our SOMs flagged a sample point as anomalous based on its MQE score, using the 99th percentile of the training MQEs as a threshold. The results are displayed in Table 2. We see that the SOMs flagged >99% and 84% in the fault test sets of Subsystem 1 and Subsystem 2 respectively. By comparison, the Subsystem 1 SOM flagged ~12% of the nominal test set as anomalous, while the Subsystem 2 SOM detected no anomalies in the nominal Subsystem 2 test set.

Table 2. Percentage Anomalies detected in Graywater Recycling System Data with 99% Confidence Interval

Subsystem	Test Dataset	Percentage Anomalies Detected
Subsystem 1	A. nominal test set B. fault test set	A. 12.2% B. 99.86%
Subsystem 2	A. nominal test set B. fault test set	A. 0% B. 84%

Subsystem 1 SOM MQE Results

Nominal Test Set

The Subsystem 1 SOM detected a relatively high percentage of anomalies in the nominal test set (~12%). This was due to anomalous behavior in the tail-end of the test set. Observe the plot comparing the MQE scores on the Subsystem 1 nominal test set (blue line) with that of the Subsystem 1 training set (orange line) in Figure 1. The MQE scores of the nominal test set spike around ~21000 data points at the end of the run. We observed that this was due to many sensors in Subsystem 1 simultaneously exhibiting low or stopped activity, likely due to a "shut down" procedure. Though this behavior is not necessarily faulty, it was not captured by the training data so represents an anomaly that we would expect our SOM to flag, as it did.

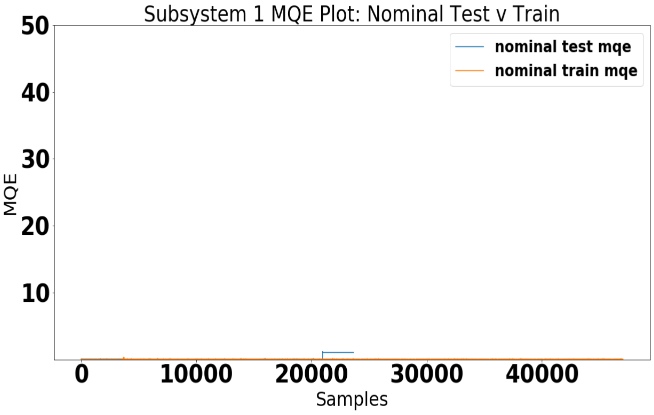

Fig. 1. Graywater Recycling Subsystem 1 Nominal Test MQE (blue) vs Train MQE (orange) Plot. Spike in nominal test MQE occurs at end of run due to shut down procedure.

Fault Test Set

Observe from Figure 2 that the MQE scores across the Subsystem 1 fault test set (blue) are significantly greater than the maximum MQE score for the Subsystem 1 training set (orange), indicating that the SOM trained on Subsystem 1 nominal data detects significant deviation in the faulty test set.

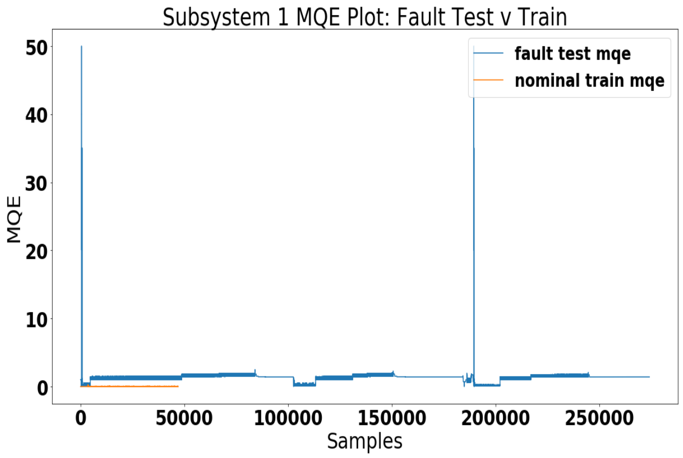

Fig. 2. Graywater Recycling Subsystem 1 Fault Test MQE (blue) vs Train MQE (orange) Plot. Fault MQE is substantially larger than nominal train MQE for entirety of run.

Subsystem 2 SOM MQE Results

The SOM trained on Subsystem 2 data did not detect any anomalies in the Subsystem 2 nominal test set. Observe in Figure 3A that this is because the MQE scores of the nominal test set closely align with the MQE scores of the training set (~0) —that is, they fall within the 99th-percentile of nominal MQE scores. By comparison, observe the significant deviation between the MQE scores of the Subsystem 2 fault set (blue) and those of the Subsystem 2 training set (orange) in Figure 3B. We see behavior similar to a stair-step function in the interval marked by [A]. In between [A] and [B], the MQE briefly drops to within nominal range before spiking again in [B]. It then oscillates between *nominal* in intervals [C] and [E] (compare with the nominal training set MQE scores, in orange) and *high* in intervals [B] and [D].

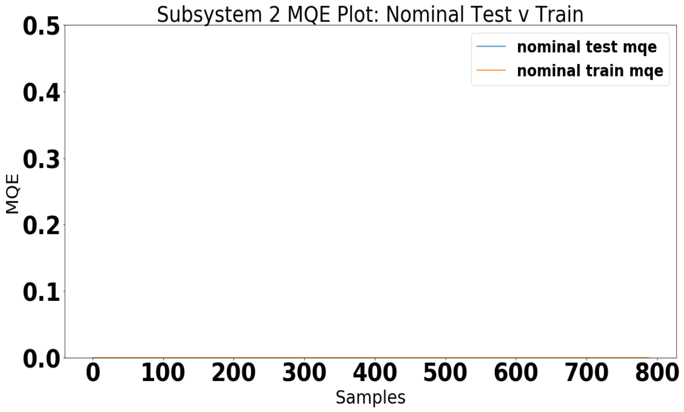

Fig. 3A. Graywater Recycling Subsystem 2 Nominal Test MQE (blue) vs Train MQE (orange) Plot. Both nominal test and train MQE's are ~0, as expected.

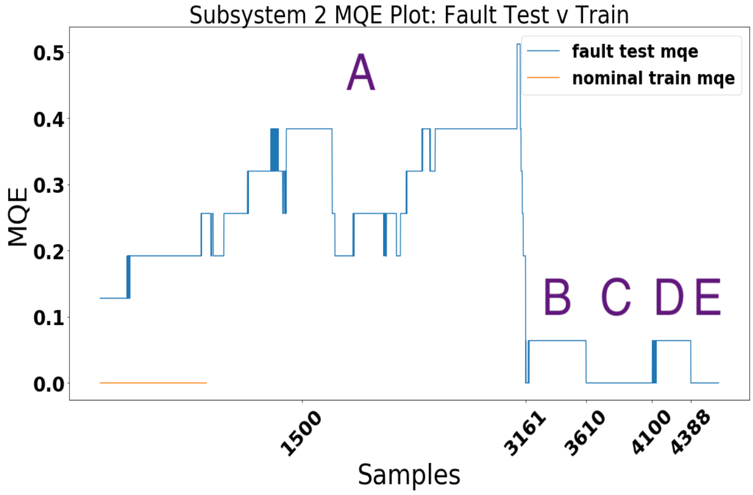

Fig. 3B. Graywater Recycling Subsystem 2 Fault Test MQE (blue) vs Train MQE (orange) Plot. Test MQE deviates significantly in intervals [A], [B], and [D].

We observed that the Subsystem 2 MQE plot closely aligns with the behavior of the CONDUCTIVITY SCALED OA measurand plotted in Figure 4, comparing the sensor values from the Subsystem 2 fault test set (in red) with those from the Subsystem 2 training set (in black). Compare Figures 3B and 4 and notice how the changes in the MQE scores in Figure 3B across intervals [A – E] correlate with the changes in behavior of the CONDUCTIVITY SCALED OA measurand in Figure 4 across the same intervals.

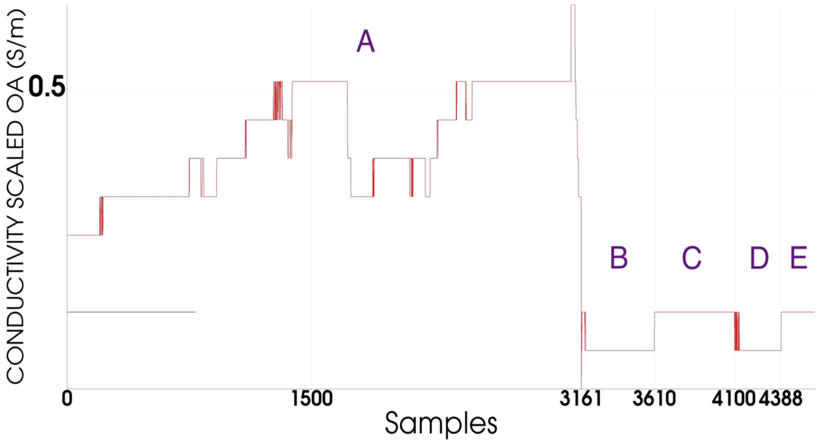

Fig. 4. Graywater Recycling Subsystem 2 CONDUCTIVITY SCALED OA: red (test), black (train). Test data deviates significantly from training in intervals [A], [B], and [D].

In particular, the CONDUCITIVITY SCALED OA measurand in Figure 4 experiences dramatic deviation from the nominal training data (in black) in a stepwise fashion throughout interval [A], while MQE scores in Figure 3B deviate from nominal range in a similar way. It then oscillates between a low reading during the intervals [B] and [D], and nominal readings in the intervals [C] and [E]. Similarly, the MQE scores in Figure 3B deviate from nominal range in intervals [B] and [D] and return to within nominal range in intervals [C] and [E].

This demonstrates the SOMs ability to not only detect deviations in the fault data, but also to capture the *relative* degree of deviation exhibited in a fault. That is, the greater the deviation from nominal behavior seen in training, the greater the MQE score will be. This is a significant innovation for an effective ISHM tool, as it distinguishes between severe (and potentially fatal) faults and more mild anomalies, allowing end-users to prioritize their overhauling and response activities accordingly.

5.3 Anomaly Localization Analysis

Our ExtraTreeClassifier (ETC) algorithm identified the sensors in Table 3 as contributing the most to the anomalies detected in each subsystem's fault test set. We only listed the measurands with a feature importance score of at least 10. We validated our results with a subject matter expert (SME) who worked at Kennedy Space Center for 32 years, including on the actual International Space Station (and other spacecraft) systems and consumables while they were on the ground.

Table 3. Salient Features of Flagged Anomalies from ETC for Graywater Recycling Subsystems

Subsystem	**Test Data**	**Salient Measurands Identified**
Subsystem 1	fault	RO PUMP SPEED: 68.6
Subsystem 2	fault	CONDUCTIVITY SCALED OA: 94.0

Subsystem 1 Validation

Our SME confirmed that the RO PUMP SPEED is one of the most important measurands to detect for a clogged FO Membrane fault, as the pump is what moves fluid from the FO Membrane through the OA Tank to the RO Membrane. When the FO Membrane is clogged by sludge, the excess build-up prevents fluid from circulating properly through the pump, so deviation from nominal pump behavior is expected and should be flagged. Such underactivity is clearly displayed in Figure 5, in which the RO PUMP SPEED from the fault test set (green) is plotted against that from the train set (black). The TURN SYSTEM ON/OFF INDICATOR is also plotted for the test set (red). Notice that the RO PUMP SPEED deviates from nominal each time the system is turned on.

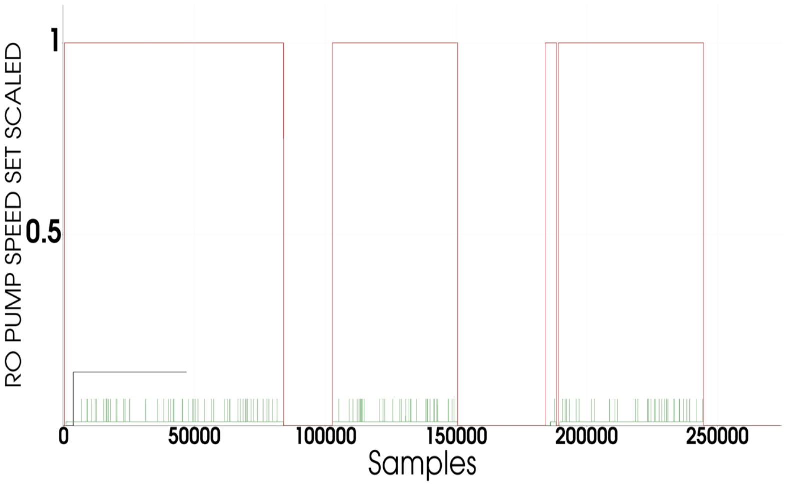

Fig. 5. Graywater Recycling Subsystem 1 RO PUMP SPEED SET SCALED: green (test), purple (train). TURN SYSTEM ON/OFF INDICTOR: red (test).

Subsystem 2 Validation

The CONDUCTIVITY SCALED OA measurand measures the electrical conductivity of the fluid moving through the recycling system. During a clog, the water contains greater mineral and salt deposits, which affects the conductivity of the fluid. Our SME confirmed that we would expect to see deviation in the CONDUCTIVITY SCALED OA measurand for the OA system, since it receives fluid directly from the clogged FO Membrane implicated in the fault. Furthermore, refer back to Figure 4 which displays the significant deviation in CONDUCTIVITY SCALED OA in the test set from training, and to Figure 3B which shows the corresponding MQE scores for the Subsystem 2 fault test set. The correlation between these two plots indicates that the CONDUCTIVITY SCALED OA was a large contributor to the deviation in the fault data. The fact that our ETC algorithm correctly identified it as the top salient feature proves the effectiveness of our fault localization approach.

6 Conclusions and Future Work

Our research demonstrates the feasibility of applying an unsupervised, SOM-based anomaly detection approach to the Integrated System Health Management (ISHM) domain for space subsystems and lays the foundation for behavior diagnosis through our anomaly localization techniques that isolate measurands most correlated with flagged anomalies. We were able to demonstrate these results on a NASA ARC Graywater Recycling System dataset implicated by a known fault. Moreover, our research makes use of Python packages that use highly parallel processing techniques to produce computationally efficient results.

As this work was a relatively small feasibility study to investigate the utility of SOM-based analysis for space subsystem health monitoring, we have relegated several important research questions to future work, which we mention here briefly. In addition to taking a more principled approach to hyperparameter tuning, e.g. through cross-validation or Bayesian optimization, we will build upon our existing methods through incorporating SOM-based prognostics capabilities based on the work of [17] and implementing multi-timescale analysis in order to cross-correlate anomalies across subsystems from data collected across different timescales.

Furthermore, we intend to investigate methods for exploiting the visualization capabilities of SOMs as in [2] for the purpose of fault localization and characterization. For instance, displaying the component planes of sensors highly correlated with data implicated in a fault may assist human operators in more quickly diagnosing and responding to flagged anomalies. We will continue discussions with NASA engineers to better understand the desirability and effectiveness of such visualizations.

While the NASA data we received did not contain confounding anomalies, nor severely unbalanced classes (e.g. 99% nominal samples versus 1% anomalous samples), effectively handling such cases is important for generalizing ADTM to new subsystems, and we have included such analysis as part of ongoing work. Finally, we intend to compare the results of our techniques to other unsupervised approaches, such as k-means and PCA, in order to further establish the utility of our approach to real applications.

7 References

1. Crusan J (2016) Habitation Module, NASA Advisory Council, Human Exploration and Operations Committee. 3.
2. Cottrell M, Gaubert P, Eloy C et al. (2009) Fault Prediction in Aircraft Engines Using Self-Organizing Maps. Advances in Self-Organizing Maps 37-44. doi: 10.1007/978-3-642-02397-2_5
3. Bennouna O, Heraud N, Leonowicz Z (2012) Condition monitoring &amp; fault diagnosis system for Offshore Wind Turbines. 2012 11th International Conference on Environment and Electrical Engineering. doi: 10.1109/eeeic.2012.6221389
4. Pascoal C, de Oliveira M, Valadas R et al. (2012) Robust feature selection and robust PCA for internet traffic anomaly detection. 2012 Proceedings IEEE INFOCOM. doi: 10.1109/infcom.2012.6195548
5. Nassar B, Hussein W, Mokhtar M (2019) Space Telemetry Anomaly Detection Based on Statistical PCA Algorithm. In: Zenodo. http://doi.org/10.5281/zenodo.1109667.
6. Gaddam S, Phoha V, Balagani K (2007) K-Means+ID3: A Novel Method for Supervised Anomaly Detection by Cascading K-Means Clustering and ID3 Decision Tree Learning Methods. IEEE Transactions on Knowledge and Data Engineering 19:345-354. doi: 10.1109/tkde.2007.44
7. Iverson D, Martin R, Schwabacher M et al. (2009) General Purpose Data-Driven System Monitoring for Space Operations. AIAA Infotech@Aerospace Conference. doi: 10.2514/6.2009-1909
8. Gao Y, Yang T, Xu M, Xing N (2012) An Unsupervised Anomaly Detection Approach for Spacecraft Based on Normal Behavior Clustering. 2012 Fifth International Conference on Intelligent Computation Technology and Automation. doi: 10.1109/icicta.2012.126

9. Datta A, Mavroidis C, Hosek M (2007) A Role of Unsupervised Clustering for Intelligent Fault Diagnosis. Volume 9: Mechanical Systems and Control, Parts A, B, and C. doi: 10.1115/imece2007-43492
10. Tian J, Azarian M, Pecht M (2014) Anomaly Detection Using Self-Organizing Maps-Based K-Nearest Neighbor Algorithm. European Conference of the Prognostics and Health Management Society 5
11. Kohonen T (1982) Self-organized formation of topologically correct feature maps. Biological Cybernetics 43:59-69. doi: 10.1007/bf00337288
12. Wittek P, Gao S, Lim I, Zhao L (2017) somoclu: An Efficient Parallel Library for Self-Organizing Maps. Journal of Statistical Software. doi: 10.18637/jss.v078.i09
13. Saranya C, Manikandan G (2013) A Study on Normalization Techniques for Privacy Preserving Data Mining. International Journal of Engineering and Technology 5:2701-2704.
14. Geurts P, Ernst D, Wehenkel L (2006) Extremely randomized trees. Machine Learning 63:3-42. doi:10.1007/s10994-006-6226-1
15. Breiman L (2001) Machine Learning 45:5-32. doi: 10.1023/a:1010933404324
16. Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. Machine Learning 46:389-422. doi: 10.1023/a:1012487302797
17. Rai A, Upadhyay S (2017) Intelligent bearing performance degradation assessment and remaining useful life prediction based on self-organising map and support vector regression. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 232:1118-1132. doi: 10.1177/0954406217700180