

A self-supervised task for fault detection in satellite multivariate time series

*Original*

A self-supervised task for fault detection in satellite multivariate time series / Cena, Carlo; Bucci, Silvia; Balossino, Alessandro; Chiaberge, Marcello. - ELETTRONICO. - (2024), pp. 292-296. (Intervento presentato al convegno The first joint European Space Agency SPAICE Conference / IAA Conference on AI in and for Space tenutosi a Harwell (UK) nel 17 - 19 September 2024) [10.5281/zenodo.13885595].

*Availability:*

This version is available at: 11583/2993064 since: 2024-10-04T14:53:00Z

*Publisher:*

Zenodo

*Published*

DOI:10.5281/zenodo.13885595

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)

# A Self-Supervised Task for Fault Detection in Satellite Multivariate Time Series

Carlo Cena<sup>\*1,2</sup>, Silvia Bucci<sup>2</sup>, Alessandro Balossino<sup>2</sup>, Marcello Chiaberge<sup>1</sup>

<sup>1</sup>Politecnico di Torino, Torino (TO), ITA

<sup>2</sup>Argotec S.R.L., San Mauro (TO), ITA

In the space sector, due to environmental conditions and restricted accessibility, robust fault detection methods are imperative for ensuring mission success and safeguarding valuable assets. This work proposes a novel approach leveraging Physics-Informed Real NVP neural networks, renowned for their ability to model complex and high-dimensional distributions, augmented with a self-supervised task based on sensors' data permutation. It focuses on enhancing fault detection within the satellite multivariate time series. The experiments involve various configurations, including pre-training with self-supervision, multi-task learning, and standalone self-supervised training. Results indicate significant performance improvements across all settings. In particular, employing only the self-supervised loss yields the best overall results, suggesting its efficacy in guiding the network to extract relevant features for fault detection. This study presents a promising direction for improving fault detection in space systems and warrants further exploration in other datasets and applications.

## 1 Introduction

Today's world is increasingly reliant on satellite technology for navigation [1], communication [2], and scientific studies [3, 4]. It is therefore important to ensure the reliability and longevity of these assets. The challenges presented by the space environment, characterized by extreme temperatures, radiation exposure, and limited, if not absent, maintenance opportunities, underscore the critical importance of fault detection in satellite systems [5, 6]. Traditional approaches often rely on predetermined rules or thresholds, which can be inadequate for capturing the complex and dynamic nature of faults in such demanding conditions, or costly model-based approaches [7, 8].

To address these challenges, in recent years we have witnessed a surge of studies in Artificial Intelligence (AI) methodologies for fault detection in space systems, which have shown promising results

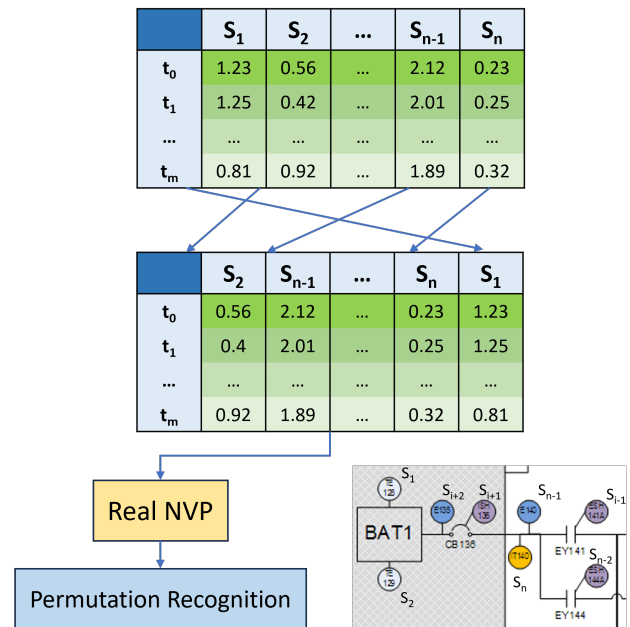


Figure 1: Self-supervised loss: the dataset's columns are permuted and the Real NVP model is trained to predict the correct permutation. As shown in the image patch extracted from the ADAPT circuit, each column is associated with a given sensor of the testbed.

in automatically identifying and diagnosing anomalies in satellite data. Among these methodologies, the Physics-Informed Real NVP model introduced by [9] demonstrated its ability to model complex distributions while incorporating domain-specific knowledge.

This model integrates principles of physics, specific to the considered dataset, ADAPT [10], into the loss function, enabling the model to capture underlying physical relationships within the data. By leveraging the Normalizing Flow family [11, 12] and affine coupling layers [13], this model excels in representing high-dimensional distributions, making it well-suited for analyzing multivariate satellite data.

Given the scarcity and cost of labeled data in this sector, self-supervision offers the advantage of lever-

\*Corresponding author. E-Mail: carlo.cena@polito.it

aging unlabeled data to guide the learning process, enhancing the model’s ability to extract meaningful features relevant to fault detection [14]. These approaches typically rely on creating pretext tasks that provide supervisory signals for model training. Their success depends on the effectiveness of the designed pretext tasks, which can be categorized into various paradigms, such as contrastive learning and generative learning [15]. Contrastive methods, like SimCLR [16], focus on learning representations by encouraging the model to distinguish between similar and dissimilar data points. Conversely, generative approaches, such as Masked Autoencoders [17], train models to reconstruct the original data from a corrupted version, forcing them to capture essential features.

Self-supervised tasks should be carefully chosen to exploit the specific structure of the data at hand and steer learning toward robust representations. Examples from other fields include object recognition across different domains using jigsaw puzzles as a self-supervised task [18] or augmentation-aware self-supervised task in the discriminator network of a GAN for robust representation learning [19].

Time series data presents unique challenges for self-supervised approaches due to their sequential nature and the importance of capturing temporal dependencies. Traditional methods primarily focus on inter-sample relationships, neglecting the crucial intra-temporal structure within a single time series.

Recent advancements address this gap by incorporating self-supervised tasks to learn inter-sample and intra-temporal relationships [20, 21]: SelfTime [20] proposes a framework that explores these relationships simultaneously. It considers separate reasoning heads within the model to analyze both the similarities between different time series and the relationships between time steps within a single series.

The challenges become even more complex when dealing with multivariate time series, where multiple interconnected variables evolve over time. Graph Neural Networks (GNNs) offer a promising solution in this domain [14], as they can capture the relationships between different variables. [22] applies transformations (permutation, scaling, etc.) on sensor data and subsequently tries to differentiate them as a training method for human activity detection.

Recognizing the potential of self-supervised learning techniques to augment existing methodologies, we propose the integration of a self-supervision task into the fault detection process for satellites’ Electrical Power System (EPS). We propose to permute the input sensor’s measurements (i.e. channels), but differently from [22], which only aimed at detecting a generic permutation among other transformations, we

ask the network to sort them by predicting the permutation’s index, similarly to what is done in [18] with image patches.

To the best of our knowledge, this is the first time that the prediction of the permutation applied to the time series features, i.e. sensor’s data, is used in the space sector as a self-supervision task, and this is the first study that performs extensive experiments to understand its effect in a multi-task setting and as a standalone loss. To summarize, the contributions of this paper are:

1. we introduce a self-supervised task for fault detection in multivariate systems, demonstrating its effectiveness in the space domain;
2. we evaluate the above-mentioned task in multiple settings, showing its effectiveness both for pre-training and in multi-task training;
3. we show that when used as a standalone loss this self-supervised task leads to better results on ADAPT [10] demonstrating its relevance especially when labels are not available.

Through these contributions, this paper aims to advance the state-of-the-art in satellite fault detection towards more resilient and reliable spacecraft.

## 2 Methodology

Here we delve into the description of the model and the losses used.

The self-supervision loss is computed by permuting the order of the dataset’s columns ( $S_1, S_2, \dots, S_n$ ) (see Figure 1), and asking the model to predict the permutation applied.

In particular, the loss function is

$$L_{\text{self\_sup}} = -\frac{1}{N} \sum_{i=1}^N \sum_{p=1}^P y_{i,p} \log \frac{\exp(F(R(x_{i,p})))}{\sum_{j=1}^P \exp(F(R(x_{i,j})))} \quad (1)$$

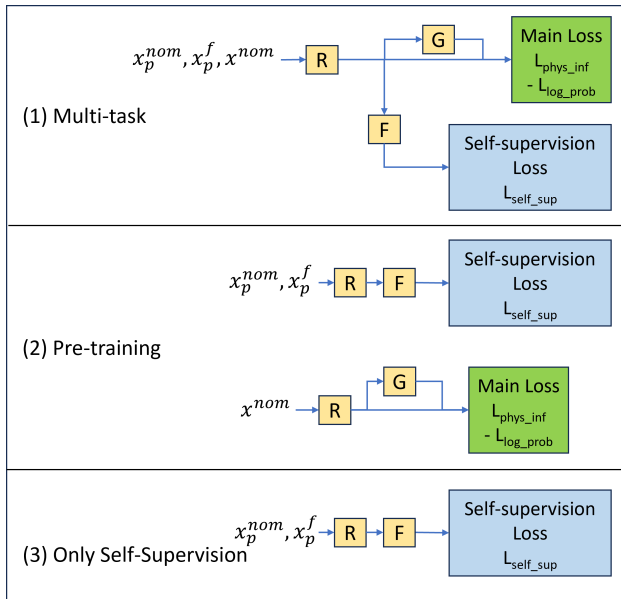
where  $N$  is the number of samples,  $P$  is the number of permutations,  $y_{i,p}$  is the label (1 if  $p$  is the correct permutation, 0 otherwise),  $x_{i,p}$  is a vector composed by concatenating 50 rows of the dataset after the application of the given permutation,  $R$  is the Real NVP model and  $F$  a fully-connected layer (see Figure 2).

We applied our loss to the configuration proposed by [9]: a Real NVP [23] neural network trained with a physics-informed loss, for fault detection.

Real NVP enables the learning of complex, high-dimensional distributions by mapping data to a simpler latent space. Unlike other Normalizing Flow models, Real NVP employs affine coupling layers to capture local dependencies and diverse modes in data,

Setting	Configuration	# Perms	AUROC	FPR95	F1	Avg. Prec.
Baseline [9]	-	-	84.92	46.81	84.51	74.64
Multi-task	-	4000	90.43	29.36	89.42	82.72
Multi-task	complete dataset	8000	90.26	35.89	89.67	83.71
Pre-Training	-	4000	90.14	30.24	89.39	82.42
Pre-Training	complete dataset	8000	92.13	26.67	90.03	83.90
Only self-supervision	-	4000	89.96	30.20	89.34	32.40
Only self-supervision	complete dataset	10000	92.50	24.13	90.41	84.68

**Table 1:** The best-performing setting is *Only self-supervision* when trained with the complete training set, with both nominal and faulty data.



**Figure 2:**  $R$ ,  $F$ , and  $G$  represent the model, a fully connected and a Gaussian distribution layer.  $p$  is the permutation applied to the input and used in the self-supervised loss, ( $L_{self\_sup}$ ). For the self-supervised task both nominal,  $x_p^{nom}$ , and fault data,  $x_p^f$ , were used, while for the main loss only nominal data,  $x^{nom}$ , was given as input. The main loss is composed by the default loss for Real NVP, ( $-L_{log\_prob}$ ), and a physics-informed loss specific to ADAPT, ( $L_{phys\_inf}$ ).

facilitating the generation of realistic samples. Each coupling layer transforms one set of variables while keeping the other unchanged through invertible functions, usually implemented as neural networks. This approach allows for efficient inference and generation of samples.

We tested our self-supervised loss by using it in three different settings, shown in Figure 2:

1. Multi-task: a multi-task training loop where both losses were used simultaneously;
2. Pre-Training: a self-supervised pre-training step followed by a fine-tuning step in which [9]’s final

loss was used, i.e. our Main Loss;

3. Only self-supervision: the self-supervision task was used as a standalone loss during training.

In all settings, to compute the self-supervision loss the architecture was augmented with a fully connected layer ( $F$ ), with a number of output units equal to the number of permutations ( $P$ ) to predict the permutation’s index among a given set through a softmax activation function.

### 3 Experimental Setup

In the following, we provide an overview of the architecture used, the dataset considered for the experiments, some implementation details, and the metrics selected to evaluate our solutions.

**Architecture:** we used the Real NVP with 4 coupling layers [13], where the neural networks used as translation functions ( $t$ ) and scale functions ( $s$ ) are made up of 2 fully-connected layers with 32 units and a final fully-connected layer with as many units as the dimension of the inputs. As activation, the  $t$  networks have a linear function, while the  $s$  neural networks have the tanh function.

**Dataset:** We used ADAPT [10], a dataset created by NASA Ames Research Center to evaluate EPS fault detection algorithms. We generated 7 splits by randomly splitting the provided files between training and testing data decreasing the number of samples by 66% with respect to those used in [9] to deal with a more challenging task and to demonstrate the efficacy of our method in conditions where labeled data is scarce and expensive. In all settings we also consider a different configuration (*complete dataset*), in which we used the complete dataset, excluding the test set of the current split, to train the self-supervision task.

**Implementation Details:** All models have been trained and evaluated on an Intel Xeon Scalable Processors Gold 6130 with 5 different random seeds on

all 7 splits using as input a time window of 50 timestamps. Each training was performed keeping 30% of the training data as validation set and using it for early stopping. Before training, the dataset was scaled between 0 and 1. The self-supervised task requires the permutation of the input features. Given the high number of possible permutations, we decided to create a few sets of permutations of different sizes as follows: for each set, given its target size  $P$ , we randomly permuted the columns for a number of times well above  $P$  and then kept the  $P$  permutations with higher entropy. Specifically, we compute

$$D = \sum_{i,j=1, i \neq j}^P \sum_{k=1}^n |i_k - j_k| + \sum_{i=1}^P \sum_{k=1}^n |k - i_k| \quad (2)$$

where  $n$  is the number of sensors present in the dataset (Figure 1),  $i$  and  $j$  indicate distinct permutations,  $i_k$  ( $j_k$ ) represents the sensor's index moved in position  $k$  by permutation  $i$  ( $j$ ). When selecting the  $P$  permutations we aim to maximize  $D$ , as this ensures greater positional deviation from the features' true positions and distinctiveness from other permutations within the set.

**Metrics:** During inference, the fully connected layer was removed, allowing for the direct use of the log-likelihood values outputted by the Real NVP model's distribution layer ( $G$ ) to evaluate the effect of the self-supervised loss. We computed the Area Under the Receiver Operating Curve (AUROC), which measures the model's ability to distinguish between classes by calculating the area under the curve generated by plotting the True Positive Rate against the False Positive Rate with various thresholds; the F1-score, which balances precision and recall by considering both false positives and false negatives; and the False Positive Rate at a 95% true positive rate (FPR95), which indicates the rate of false alarms when the true positive rate is high.

## 4 Results

In Table 1 we show, for each setting and configuration, the results obtained evaluating the model on the 7 test sets with 5 different random seeds.

All the experiments shown in Table 1 performed better than the baseline, i.e. the neural network and loss proposed by [9] showing that the self-supervised task is useful in driving the network toward the relevant features.

Additionally, the results show that in all settings the *complete dataset* configuration leads to better results in all metrics, but FPR95 in *multi-task*. This is expected as more data is used. More interesting are the results

obtained when training using only the self-supervised loss with the complete dataset, which leads to the best results overall. We believe this is because, differently from the final loss in [9], this loss uses both nominal and faulty data, successfully learning representative features about sensor correlations from both.

Moreover, the experiments showed that the use of a bigger dataset for the self-supervised task always led to an increase in the number of permutations needed for the best results. This may be due to the hardness of the chosen task, which requires a high number of samples to be successfully learned in all cases.

## 5 Discussion

We present a self-supervised task based on feature permutations to pre-train a Physics-Informed - Real NVP neural network for fault detection in multivariate time series. The experiments are performed on several custom splits of the ADAPT dataset.

The results show that using this self-supervised loss leads to improvements with respect to those that can be obtained with previous studies when dealing with a small dataset. This demonstrates the contribution of this work towards more data-efficient AI models, a particularly relevant feature in the space sector, due to the complexities involved in creating a fault detection dataset. Moreover, the proposed task has also been shown to be useful in multi-task settings, as well as a standalone loss.

We believe that the last case is particularly interesting, and we plan on further investigating it to understand the differences between the features extracted by the Real NVP model trained with it and those obtained from the models trained with the log probability and the physics-informed loss. Moreover, we plan on further analyzing the effect of the number of permutations on the self-supervision task in all settings. Finally, the results show that there may be room for additional improvements by using a higher number of samples to pre-train models using bigger sets of possible permutations.

## ACKNOWLEDGMENT

This work has been developed with the contribution of the Politecnico di Torino Interdepartmental Centre for Service Robotics (PIC4SeR <https://pic4ser.polito.it>) and Argotec SRL. This publication is part of the project PNRR-NGEU which has received funding from the MUR - DM 117/2023. Computational resources were provided by HPC@POLITO (<http://hpc.polito.it>).

## References

1. Hofmann-Wellenhof, B., Lichtenegger, H. & Wasle, E. *GNSS—global navigation satellite systems: GPS, GLONASS, Galileo, and more* (Springer Science & Business Media, 2007).
2. Rahmat-Samii, Y. & Densmore, A. C. Technology trends and challenges of antennas for satellite communication systems. *IEEE Transactions on Antennas and Propagation* **63**, 1191–1204 (2014).
3. Martini, M., Mazzia, V., Khaliq, A. & Chiaberge, M. Domain-adversarial training of self-attention-based networks for land cover classification using multi-temporal Sentinel-2 satellite imagery. *Remote Sensing* **13**, 2564 (2021).
4. Dotto, E. & Zinzi, A. Impact observations of asteroid Dimorphos via Light Italian CubeSat for imaging of asteroids LICIA-Cube. *Nature Communications* **14**, 3055 (May 2023).
5. Hasan, M. N., Haris, M. & Qin, S. Fault-tolerant spacecraft attitude control: A critical assessment. *Progress in Aerospace Sciences* **130**, 100806. ISSN: 0376-0421 (2022).
6. Xu, Y., Kong, X., Ma, W., Zhu, Z. & Zhang, J. *Integrated System Health Management Research and Application on Satellite in Global Reliability and Prognostics and Health Management (PHM-Nanjing)* (2021), 1–5.
7. Schein, J., Bushby, S. T., Castro, N. S. & House, J. M. A rule-based fault detection method for air handling units. *Energy and buildings* **38**, 1485–1492 (2006).
8. Jiang, T., Khorasani, K. & Tafazoli, S. Parameter Estimation-Based Fault Detection, Isolation and Recovery for Nonlinear Satellite Models. *IEEE Transactions on Control Systems Technology* **16**, 799–808 (2008).
9. Cena, C., Albertin, U., Martini, M., Bucci, S. & Chiaberge, M. *Physics-Informed Real NVP for Satellite Power System Fault Detection in IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)* (2024). Forthcoming.
10. Mengshoel, O. J. *et al. Diagnosing Faults in Electrical Power Systems of Spacecraft and Aircraft in Proceedings of the Twenty-Third Conference on Artificial Intelligence* (eds Fox, D. & Gomes, C. P.) (AAAI Press, July 2008), 1699–1705.
11. Tabak, E. G. & Vanden-Eijnden, E. Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences* **8**, 217–233 (2010).
12. Kobzyev, I., Prince, S. J. & Brubaker, M. A. Normalizing Flows: An Introduction and Review of Current Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **43**, 3964–3979 (2021).
13. Dinh, L., Krueger, D. & Bengio, Y. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516* (2014).
14. Zhang, K. *et al.* Self-supervised learning for time series analysis: Taxonomy, progress, and prospects. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).
15. Liu, Z., Alavi, A., Li, M. & Zhang, X. Self-Supervised Learning for Time Series: Contrastive or Generative? *arXiv preprint arXiv:2403.09809* (2024).
16. Chen, T., Kornblith, S., Norouzi, M. & Hinton, G. *A Simple Framework for Contrastive Learning of Visual Representations in Proceedings of the 37th International Conference on Machine Learning* **119** (PMLR, July 2020), 1597–1607.
17. Zhang, C., Zhang, C., Song, J., Yi, J. S. K. & Kweon, I. S. *A Survey on Masked Autoencoder for Visual Self-supervised Learning in Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI)* (International Joint Conferences on Artificial Intelligence Organization, Aug. 2023), 6805–6813.
18. Carlucci, F. M., D’Innocente, A., Bucci, S., Caputo, B. & Tommasi, T. *Domain generalization by solving jigsaw puzzles in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), 2229–2238.
19. Hou, L. *et al.* *Augmentation-Aware Self-Supervision for Data-Efficient GAN Training in Advances in Neural Information Processing Systems (NeurIPS)* **36** (Curran Associates, Inc., 2023), 31601–31620.
20. Fan, H., Zhang, F. & Gao, Y. Self-supervised time series representation learning by inter-intra relational reasoning. *arXiv preprint arXiv:2011.13548* (2020).
21. Sarkar, P. & Etemad, A. *Self-supervised learning for ecg-based emotion recognition in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2020), 3217–3221.
22. Saeed, A., Ozcelebi, T. & Lukkien, J. *Multi-task self-supervised learning for human activity detection in* **3** (ACM New York, NY, USA, 2019), 1–30.
23. Dinh, L., Sohl-Dickstein, J. & Bengio, S. *Density estimation using Real NVP in 5th International Conference on Learning Representations (ICLR)* (OpenReview.net, Apr. 2017).