Anomaly Detection in Spacecraft Telemetry: Forecasting vs. Classification

Daniel Lakey
IU International University of Applied Sciences
Germany
daniel.lakey@iu-study.org
ORCID:0000-0002-8198-7892

Tim Schlippe
IU International University of Applied Sciences
Germany
tim.schlippe@iu.org
ORCID:0000-0002-9462-8610

***Abstract*—Anomaly detection in spacecraft telemetry is critical for the success and safety of space missions. Traditional methods often rely on forecasting and threshold techniques to identify anomalies [1]–[5]. This paper presents a comprehensive comparison of traditional forecast-based anomaly detection against two innovative classification methods, including a direct classification and an image classification through Gramian Angular Field (GAF) transforms [6], which have only been analysed in other domains but not for spacecraft anomaly detection. All our investigated systems leverage deep learning architectures and use the popular real SMAP/MSL spacecraft data from [2]. Our findings suggest that direct classification provides a marginal but statistically significant improvement in anomaly detection over traditional methods. However, image classification, while less successful, offers promising directions for future research. The study aims to guide the selection of appropriate anomaly detection techniques for spacecraft telemetry and contribute to the advancement of automated monitoring systems in space missions.**

***Index Terms*—anomaly detection, time series classification, image classification**

I. Introduction

The enduring challenge to maintain the operational integrity of spacecraft hinges on the timely and precise detection of anomalies within their complex systems [1]–[5]. As missions grow in duration and complexity, traditional methods of monitoring have met with "intractable" challenges [7] that necessitate innovative approaches, particularly with regard to spacecraft anomaly detection [1]. Our work delves into the domain of anomaly detection in spacecraft time series data, an area undergoing active study and pivotal to the advancement of space mission safety and efficiency [8].

With a focus on deep learning architectures, our research embarks on a comparative analysis of anomaly detection in spacecraft telemetry, evaluating their efficacy and paving the way for a paradigm shift from conventional *forecasting & threshold*-based systems [1]–[5] to more nuanced, context-aware techniques whereby the deep learning system can identify the *anomalous* cases in a direct time series classification (*direct classification*) without relying on an operator-defined threshold. While *direct classification* for anomaly detection is an active area of research in other domains [9], [10], there has not been an analysis of *direct classification* for anomaly detection in spacecraft telemetry.

Additionally, our exploration of image transform techniques through GAF transforms [6] on the time series data of spacecraft telemetry (*image classification*) presents a novel intersection of methodologies with the potential to redefine anomaly detection strategies in spacecraft systems by extending work on GAFs for anomaly detection in other domains [11], [12].

We build on our previous paper [13], in which we compared 13 deep learning architectures when used for spacecraft anomaly detection using *forecasting & threshold* against a benchmark dataset provided by [2]. Now we extend the investigation by comparing the *forecasting & threshold* results of [13] against the corresponding results from *direct classification* and *image classification*. We share our code with the research community in our GitHub repository[1].

II. Related work

This section discusses related work and concepts upon which we base our study. In particular, the different approaches to anomaly detection are outlined and discussed.

A. Data for spacecraft anomaly detection

Modern spacecraft have many thousands of telemetry channels[2] [4], and this "huge" [14] amount of data is more than can be monitored by human operators. Within these channels, actual instances of anomalies are rare. By design a spacecraft is a robust machine, fault tolerant and extensively tested to ensure that anomalies do not occur [15]. For example, a study of seven different spacecraft over more than a decade yielded fewer than 200 critical anomalies [16].

Spacecraft anomaly detection is a particularly challenging field due to the sparsity of publicly available datasets for training. Indeed, of all the studies listed in our work, only [2] make the data available, and even then with implementation-specific details hidden through scaling and normalisation. This has led to their dataset becoming a benchmark for further studies, such as [4], [13], [17], [18]. We note there are some well-documented drawbacks with the dataset [19], but as the

[1]https://github.com/E-Penguin/SpacecraftAnomalyClassification

[2]Terminology: a *telemetry channel* consists of one or more *telemetry parameters* containing information about a spacecraft system. The *value* of a *parameter* at a given time is a *sample*. These *samples* are taken as *data points* by the deep learning models.

benchmark for some many other works, it will be acceptable given the comparative nature of this study. Consequently, we also used the "SMAP/MSL" dataset from [2] in our experiments.

The SMAP/MSL dataset comprises of 82 telemetry channels taken from the Soil Moisture Active Passive (SMAP) [20] spacecraft and "Curiosity" Mars Science Laboratory (MSL) [21] spacecraft. The data has been scaled from between (–1,1) and "*Channel IDs are also anonymized, but the first letter gives indicates the type of channel (P = power, R = radiation, etc.). Model input data also includes one-hot encoded information about commands that were sent or received by specific spacecraft modules in a given time window*"[3]. This results in a collection of 82 multivariate telemetry channels, with around 100 labelled anomalies in total across all channels. Each telemetry channel is a multivariate time series of one *target* parameter and additional parameters to be used as contextual information. The value of the target parameter is the time series to be forecast, in which anomalies are to be detected.

As part of the anonymization performed on the SMAP/MSL dataset, the timing information has been removed by the authors. It is therefore unknown what period is represented by the given data. It is not possible to overlay or combine multiple telemetry channels as we cannot assume any two telemetry channel share a comparable time base.

B. Forcasting & threshold

Forecasting & threshold is one of the most common approaches for anomaly detection. As illustrated in Fig. 1, the model forecasts a number of time steps based on the previous timesteps and learned model. The forecasted values are compared to observed values to determine how *anomalous* the observed values are. This requires the model to learn *normal* values by training (usually semi-supervised) on *normal* observed data. Anomalies are then identified by the distance between the forecast values differing from the actual values by some threshold. AutoRegressive Integrated Moving Average (ARIMA) is a statistical model used for forecasting time series data, combining autoregressive (AR), differencing (I), and moving average (MA) components to capture various temporal structures and trends in the data [22]. Telemanom [2] and DeepAnT [23] are both deep learning forecasters, using Long Short Term Memories (LSTM) and Convolutional Neural Networks (CNN) respectively, to learn the normal behaviour of the data. As multivariate models they are particularly suited to spacecraft anomaly detection.

In Fig. 1, the blue plot "Actual Data" represents one parameter within a telemetry channel, the input data. The orange plot "Forecast Data" is the prediction output of some regression model, previously trained on *normal* data. Where there is a divergence between the two values at a given time point, greater than the defined threshold, an error is determined and an anomaly declared. Superimposed on the plot is an indication of anomaly sequences containing multiple error points (collective anomalies), and the error threshold above and below the forecast values.

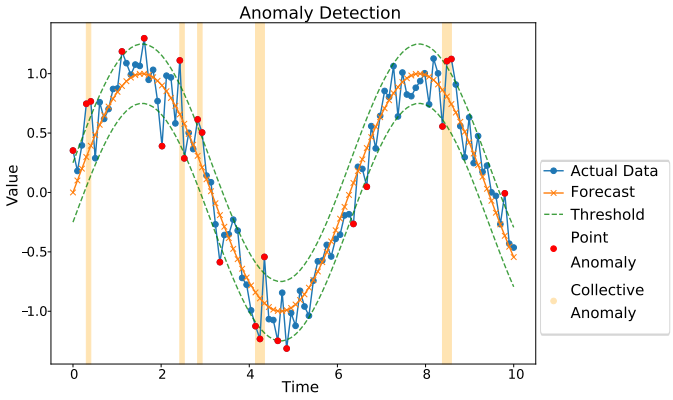


Fig. 1: Anomaly detection with *forcasting & thresold*

Whilst effective, *forecasting & threshold* relies on the selection of some threshold value beyond which the construction error is considered *anomalous*. For example, [2] propose a "unsupervised and nonparametric anomaly thresholding approach" where the anomaly detector dynamically learns the error value corresponding to "anomaly" for a particular time series. In their work, for each telemetry channel, they train a different *forecasting & threshold* model.

There have been very many studies on a wide range of models for time series anomaly detection. [24] mention 158 models as of 2022; no doubt the numbers have increased since. A comprehensive comparison of approaches is given in [25]. Over half of the models considered in [25] are for *univariate* data, in spacecraft telemetry terms: single parameter. As shown earlier, spacecraft data is tightly coupled with context and correlation between parameters—a *multivariate* approach is required. This excludes some popular statistical-based time series forecasting methods like ARIMA [26].

C. Direct classification

While *forecasting & threshold* and related approaches have their merits, it often requires manual tuning of thresholds and assumes a certain structure in the data, which may not always hold true. Furthermore, it relies on the accuracy of the forecasting model, where any imprecision can lead to false alarms or missed anomalies. Rather than simply learning the normal data behaviour, the deep learning model should be able to learn the implied thresholds also, without the manual tuning that threshold-based systems generally require.

Instead of forecasting future values, *direct classification* employs classifiers to label each time point or window of the series directly as *normal* or *anomalous*. With the advent of deep learning, models like CNNs and LSTMs have shown significant promise in handling time series data with classification [27]–[29]. These models capture intricate patterns and dependencies in the data without assuming any explicit structure, offering a more flexible and often more accurate alternative to

[3]https://github.com/akshu281/KDD_LSTM

the traditional approach. By using *direct classification*, one can bypass the potential pitfalls of forecasting inaccuracies and threshold tuning [30]. Capturing the temporal dynamics of time series is a challenging task [31]. But studies have shown that is it possible for CNN [32] and recurrent neural network (RNN) [33] classifiers. The use of *direct classification* specifically for anomaly detection is an area of active research, such as [28], [34]. Our study is, to the best of our knowledge, the first to present it in the context of *telemetry* anomalies, that is, of the time series data reporting the spacecraft health.

D. Image classification

Image classification in the context of anomaly detection means that the time series data is first transformed into an image and then an image classification is applied to classify the time series expressed as image into *normal* or *anomalous*.

Neural networks, particularly CNNs, perform well at image recognition [35], especially in the anomaly detection domain [36]. [6] and [37] propose a novel method for *image classification* of time series data. A variety of transformations are examined in [38]. The particular transformation [6], [37], [38] applied is the Gramian Angular Field (GAF) [39]. [40] presents a detailed and thorough demonstration of GAF's core concepts and mathematical underpinnings.

GAF has two innovations [37]: Firstly, the 1-dimensional time series data is encoded into two dimensions through projection into a polar coordinate frame. Essentially, the time is now expressed as the radius r whilst angle θ represents the value. The second innovation is the use of the Gram matrix to preserve "*the temporal dependency. Since time increases as the position moves from top-left to bottom-right, the time dimension is encoded into the geometry of the matrix*" [40]. Thus, the time series is encoded in a 2-dimensional image. Fig. 2 demonstrates the process with simulated data, showing GAFs for *normal* and *anomalous* time series. Fig. 2a shows how time series without anomalies are encoded into polar representation, and from there to a GAF. Fig. 2b illustrates the effect of a time series anomaly on the computed GAF, visible primarily on the bottom-left corner of the transformed image as a discontinuity in the pattern.

[41] applied the GAF transformation to financial forecasting, a time series problem, whereas [42] demonstrated the use of the GAF transformation to perform anomaly detection. To the best of our knowledge, we are the first to apply GAFs in the domain of spacecraft anomaly detection, such that it can classify *anomalous* time series.

III. Experimental Setup

This section details the implementation of our experiments to compare the performance of *direct classification* and *image classification* against the *forecasting & threshold* approach which performed best in [13].

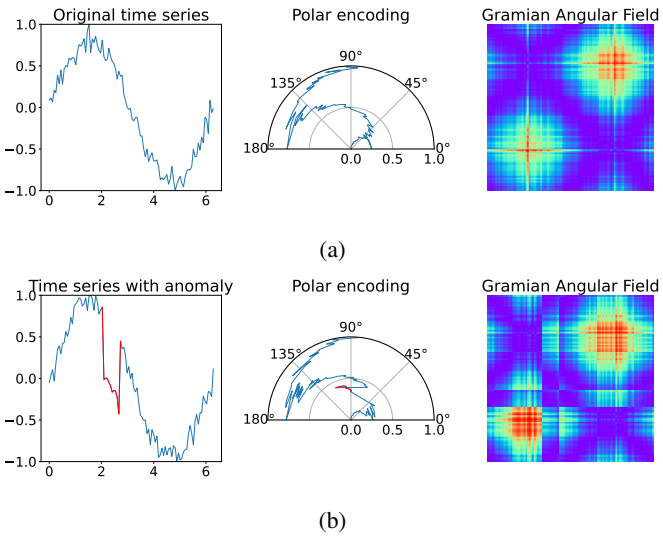


Fig. 2: GAFs for *normal* and *anomalous* time series

A. Data preparation

As described in section II-A, we used the SMAP/MSL dataset[4] from [2] to train and test *forecasting & threshold*, *direct classification* and *image classification* systems for anomaly detection in spacecraft telemetry. The dataset is extremely imbalanced. Considering only the data partition containing anomalies, the proportion of *anomaly* vs. *normal* data points ranges from 0.4% to 50%, with a median value of around 6%. It should also be borne in mind that this is a *curated* dataset which has been focussed on known anomalies, whereas in a real-world use case there could be years of *normal* data preceding an *anomaly* occurrence. A 6% anomaly rate over the course of a year would represent nearly 22 days of anomalies, which is far worse than generally observed [16].

Unfortunately, the default splits from [2] are not suitable for the classification experiments—the "train" data partition contains no anomalies from which to learn both *normal* and *anomalous* classes. Therefore, the original "test" partition, containing anomalies, needs to be used but split such that there are a (roughly) even number of anomalies in each set. We performed the split by finding the mid-point between anomalies such that an even number of anomalies were on either side of the split point. Not all telemetry channels contained more than one anomaly (one contained none at all), resulting in the number of telemetry channels available for the classification experiments being reduced. Table I details the telemetry channels remaining after splitting (channels with fewer than two anomalies were discarded), along with the dimensions of the datasets as N parameters $\times$ M samples. Ultimately, 10 channels remain for SMAP and 6 for MSL, the majority with just one anomaly in each partitioned dataset.

No channel has more than two anomalies in the training dataset, while all channels have exactly one anomaly in the

[4]https://s3-us-west-2.amazonaws.com/telemanom/data.zip

test dataset (Table I). This represents quite a challenge for a classifier to learn. However, in the context of spacecraft data this is not unusual—anomalies are by their nature rare: "*1–10 anomalies reported per spacecraft per year*" reported by [43].

Image classification and *direct classification* rely on "windowed" time series data, that is, they classify discrete sequences rather than individual data points. Correct selection of the window size is critical to capturing the underlying patterns and dynamics of the data [44]–[46]. Across all telemetry channels, the average anomaly length is around 240 samples. Initial investigations suggested that a window size of 64, approximately one quarter of the average anomaly size, was suitable for both *direct classification* and *image classification* experiments. Table I also shows the number of windows per telemetry channel, with a window length of 64 samples.

TABLE I: Number of anomalies, samples and windows in training and test datasets

Channel	Training			Test		
	Anom.	Samp.	Wind.	Anom.	Samp.	Wind.
SMAP						
E-1	1	26 × 5320	84	1	26 × 3196	50
E-12	1	26 × 5330	84	1	26 × 3182	50
E-11	1	26 × 5332	84	1	26 × 3182	50
E-10	1	26 × 5325	84	1	26 × 3180	50
E-13	2	26 × 6044	95	1	26 × 2596	41
G-7	2	26 × 6330	99	1	26 × 1699	27
P-1	2	26 × 4157	65	1	26 × 4348	68
P-4	2	26 × 3560	56	1	26 × 4223	66
T-1	1	26 × 5224	82	1	26 × 3388	53
T-3	1	26 × 3690	58	1	26 × 4889	77
Total	**14**	**26 × 50312**	**791**	**10**	**26 × 33883**	**532**
MSL						
C-1	1	56 × 1425	23	1	56 × 839	14
C-2	1	56 × 965	16	1	56 × 1086	17
F-7	2	56 × 3057	48	1	56 × 1997	32
P-11	1	56 × 1561	25	1	56 × 1974	31
T-13	1	56 × 1345	22	1	56 × 1085	17
T-9	1	56 × 850	14	1	56 × 246	4
Total	**9**	**56 × 12058**	**193**	**8**	**56 × 8516**	**137**

B. Evaluation Metrics

Evaluation metrics frequently employed to measure anomaly detection systems' performance capabilities are the number of False Positive (FP) ("events" incorrectly detected as *anomalous* that were actually *normal*), False Negative (FN) (the *anomalous* parts of the signal incorrectly annotated as *normal* by the algorithm), and True Positive (TP) (the *anomalous* instances correctly detected by the algorithm) that are recorded. True negatives are not reported, as they do not form a part of the F1 calculation and are not relevant to a *per-anomaly* metric.

Temporal features of time series in which anomalies occur are not recognised by such conventional per-sample measures as F-score [47], [48]. Most anomalies, especially the majority of anomalies in satellite telemetry, are often continuous sequences of correlated data. A full treatment of these metrics, and potential pitfalls, are given in [49] and [50]. The Telemanom study by [2] adopts the approach of "*Per-anomaly Precision and Recall*", and subsequently *per-anomaly F1 score*. This is illustrated in Fig. 3.

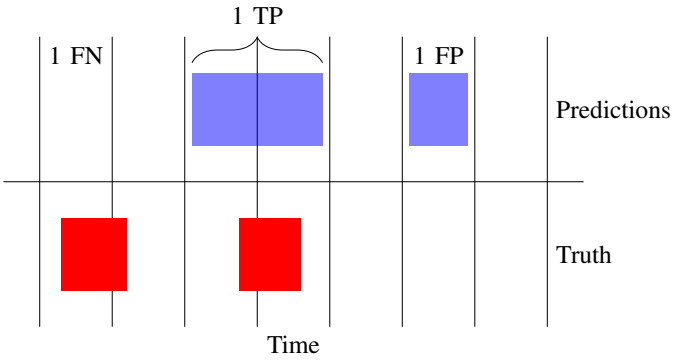


Fig. 3: Per-anomaly scoring

The concept is to treat the whole anomaly range as a single instance rather than counting each data point (*sample*) separately. That is, if at least one sample from the predicted sequence overlaps at all with the true anomaly sequence, the whole predicted range is considered part of the single TP. This is particularly important when using a windowed approach, as a true anomaly can span multiple windows. Fig. 3 show three cases: FN when there is a true anomaly but no corresponding detection; TP when the predicted anomaly range overlaps in some way with a true anomaly, and FP when the prediction does not overlap with any *anomalous* samples. Neighboring predictions are merged into a single prediction.

This concept has the advantage of being relatively simple to implement and requires no tuning. Furthermore, as we used the SMAP/MSL dataset from [2], we also adopted the same *per anomaly* evaluation metrics (TP, FP, FN, F1 score) for our experiments.

A flaw with the *per-anomaly F1 score* metric is that if a model simply predicts *every* data point as an anomaly (*all-anomaly prediction*), then it would be counted as a single TP, leading to a 100% F1 score, illustrated in Fig. 4. This affected two channels in the *image classification* experiments. Consequently, we excluded their scores from the results.

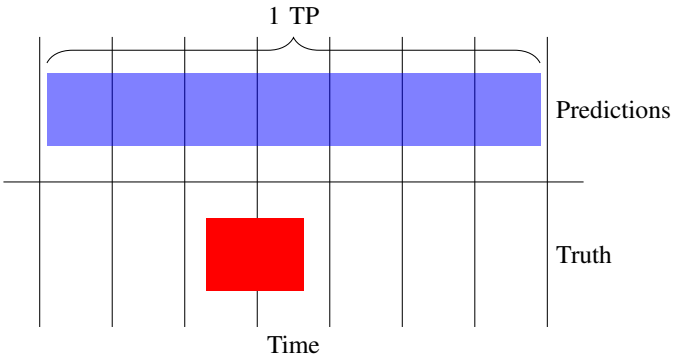


Fig. 4: *Per-anomaly* scoring with *all-anomaly prediction*

C. Forecasting and threshold

In our previous paper [13], we extended the "Telemanom" implementation[5] of [2], which relied on an LSTM-based model to (1) provide the forecasted data points and (2) assess those data points for being *anomalous* by a novel non-parametric thresholding mechanism. We analysed 13 deep learning architectures for *forecasting & threshold*, replacing the LSTM-based forecaster from [2], while retaining the thresholding mechanism. The best performing model was XceptionTimePlus, a CNN-based model originally implemented as XceptionTime [51] from the `tsai` time series analysis framework [52]. Following the approach of [2], we trained one model per telemetry channel (*one-model-per-channel* approach).

In this study, we evaluate the best performing *forecasting & threshold* system from our previous paper [13] on our new test set, which is shown in Table I, and compare it to the best *direct classification* and *image classification* systems. Again, we followed the *one-model-per-channel* approach.

D. Direct classification

For the implementation of *direct classification* we used XceptionTimePlus, the `tsai` [52] implementation of the XceptionTime model [51], with the parameters that were optimal in our previous study on the SMAP/MSL dataset [13]. XceptionTime has also been shown to be the best performing model in other (non-spacecraft related) anomaly detection studies [53], [54] and is able to deal with the multivariate telemetry channels of the SMAP/MSL dataset, thus it is a good fit for our study. We created and ran *direct classification* within the Python time series classification framework `tsai` [52], running within a Google Colab environment [55] configured to use a T4 GPU [56].

Furthermore, we elaborated a classification pipeline (Fig. 5) that instantiates the selected loss function, optimiser and weights, creates the deep learning model, and partitions the datasets into windows according to the provided window size parameters, as discussed in Section III-A. This performs the classification and returns the results (TP, FP, FN, F1 score). In addition to the *direct classification* experiments, we used this pipeline for the *image classification* experiments having a "transform" stage added.

E. Image classification

Image encoding of the time series data has the potential to perform well in a classification task, by using image classifiers after transforming the time series data into images [27], [38]. We use the `pyts` [57] framework to perform the GAF transformation, as part of the pipeline described in Section III-D. This transformation step is represented by the green box in Fig. 5.

The GAF transformation process described in Section II-D is suitable only for *univariate* (single parameter) data, whereas

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

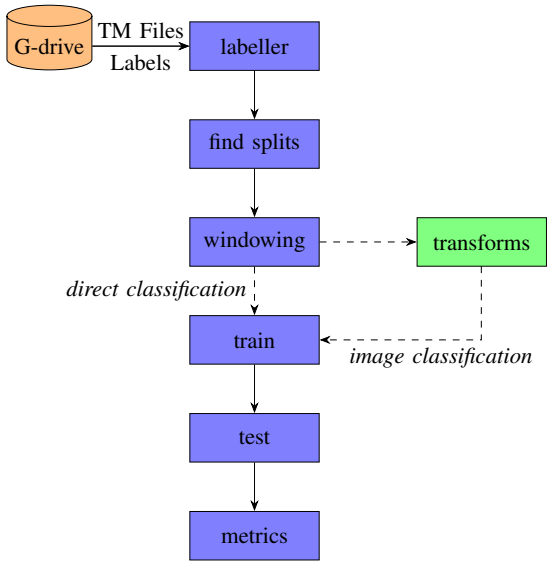


Fig. 5: *Direct* and *image classification* experiment pipeline

the SMAP/MSL dataset, and spacecraft telemetry in general, is predominantly *multivariate* (multiple parameters). The relationships between these multiple parameters defines the context of the spacecraft and is critical to understanding the *anomalous* behaviour against the backdrop of other contemporaneous spacecraft activity. Consequently, the transformer creates a "stack" of GAF images (such as in Fig. 2), one per spacecraft parameter, which is then fed through the rest of the classification pipeline providing a $224 \times 224 \times x$ image, where x is the number of parameters within the data channel. We note that `pyts` does support a multivariate image transform, based on "recurrence plots" [58]. However, our initial experiments with this approach did not produce good results (zero anomaly detections) so was dropped in favour of the GAF pipeline described above.

The deep learning model XceptionTimePlus which we use for *direct classification*, as described in Section III-D, is not suited to *image classification* as the number of dimensions is greater than the 1-dimensional time series the models accept (images being 2-dimensional). However, the `tsai` framework provides an implementation of the CNN-based ResNet34 [59], [60] called xResNet34 which is better suited to an *image classification* task. This *image classification* model provides state-of-the-art performance and is used in many recent works such as [61]–[63].

We ran the *image classification* experiments starting with an untrained xResNet34 model and pre-trained xResNet34 model from the `PyTorch` [64] collection. We trained the untrained model and fine-tuned the pre-trained model using the pipeline described in Fig. 5. To differentiate the trained and fine-tuned models, the latter is styled "xResNet34$_{\text{(fine-tuned)}}$". The number of samples available (given in Table I) is barely sufficient for the training of the untrained model [65], so we anticipated that the fine-tuned model would be superior.

We chose 224×224 for the transformed image size for

two reasons: Firstly, this is a standard image size in the literature [66]–[68], and has been shown to generally perform well compared to larger or smaller images [69]. Secondly, it is the resolution on which the xResNet34$_{\text{(fine-tuned)}}$ model is originally trained [70].

IV. Results

In this section, we will present our results of the experiments in time series anomaly classification. We will report the performances of *forecasting & threshold*, *direct classification*, and *image classification*, considering the *per-anomaly F1 score*, TP, FP and FN, as described in Section III-B.

To enable a fair comparison taking into account that the telemetry channels are of completely different lengths, we do not report the average of the individual telemetry channels' F1 scores as *overall* F1 scores for the SMAP spacecraft and MSL spacecraft and the *total* F1 score for the complete SMAP/MSL dataset in Table II and Table III, as is done in other machine learning use cases with balanced data. In contrast, we computed the *overall* and *total* F1-score based on the total count of TP, FP, and FN of all corresponding telemetry channels, reporting a generalised behaviour of the anomaly detection approaches.

A. Forcasting & threshold vs. direct classification

Table II compares the performances of *forecasting & threshold* and *direct classification* across the channels of the SMAP and MSL spacecraft. Overall, the F1 score of *direct classification* is 4% (relative) higher than the F1 score of *forecasting & threshold* (54.5% vs. 52.2%). The Wilcoxon Signed-Rank test [71] indicates that there is a significant large difference between *forecasting & threshold* (*Mdn*=0, *n*=16) and *direct classification* (*Mdn*=100, *n*=16), where Z=2.8, p=0.005, r=0.9.

However, the number of TP (correctly detected anomalies) is 50% (relative) higher than with *forecasting & threshold* (9 vs. 6). The number of FP (falsely detected anomalies) is 11 times greater (11 vs. 1) with *direct classification*.

Comparing the F1 scores of SMAP and MSL, we see that *forecasting & threshold* outperforms by 18% (relative) for SMAP (70.6% vs. 60.0%), whereas the reverse is true for MSL, where only *direct classification* (48%) was able to detect any anomalies.

B. Image classification: Re-training vs. fine-tuning

As shown in Table III, overall the *image classification* performed poorly in terms of F1 score for both xResNet34 and xResNet34$_{\text{(fine-tuned)}}$ (19.5%, 26.3%), with a high number of FP (86, 68) and lower number of TP (11, 13). The relative difference between xResNet34 and xResNet34$_{\text{(fine-tuned)}}$ is 35%. The number of samples with non-zero differences (3) is too small to perform a Wilcoxon Signed-Rank test, so we performed instead a Sign Test [72]; the z-value is 1, the p-value is 0.3. The result is not significant at $p < 0.05$.

This shows that differences between xResNet34 and xResNet34$_{\text{(fine-tuned)}}$ were negligible, which was unexpected. However, looking at the general number of TP and FP as well as the F1 scores of SMAP and MSL demonstrates that xResNet34$_{\text{(fine-tuned)}}$ performs better. Only the general number of FN is the same in both models. Consequently, we used xResNet34$_{\text{(fine-tuned)}}$ for comparison to *direct classification* in the next Section IV-C.

C. Direct classification vs. image classification

The total F1 score of *image classification* with xResNet34$_{\text{(fine-tuned)}}$ (26.3%) is around half that of *direct classification* (54.5%). *Direct classification* produces a slightly greater number of TP (15 vs. 13), a smaller number of FP (24 vs. 68), and a 5 times lower number of FN (1 vs. 5). The Wilcoxon Signed-Rank test indicates that there is a significant large difference between *direct classification* (Mdn=100, n=16) and *image classification* (Mdn=25.4, n=16), where Z=–2, p=0.050, r=–0.5.

For MSL, only 4 channels are better predicted in terms of F1 score with *image classification*: C-2, F-7, P-11, T-13. No telemetry channels are better predicted for SMAP with *image classification*. However, *direct classification* shows nearly three times fewer FP (24 vs. 68) than *image classification*.

V. Discussion

This section discusses the suitability of our novel *direct classification* and *image classification* approaches for spacecraft anomaly detection, and difficulties encountered.

A. Suitability for spacecraft anomaly detection

Following the approach of [2], throughout this work we have trained one model per telemetry channel. Modern deep learning models are expensive to train, in terms of computation. Some of the models have taken multiple hours to train to completion (Section V-D), which limits the amount of progress that can be made when iterating through different implementations and approaches.

Image classification particularly brings a large overhead to the processing time. For practical reasons the image transforms are performed "on the fly" as part of the data processing pipeline, window by window. This reflects the "stream" nature of spacecraft data. In an operational use case, such an anomaly detector would be expected to analyse the incoming spacecraft telemetry in a reasonable time to produce actionable results. This does not necessarily have to happen in real time, but should be considered in a real-world implementation.

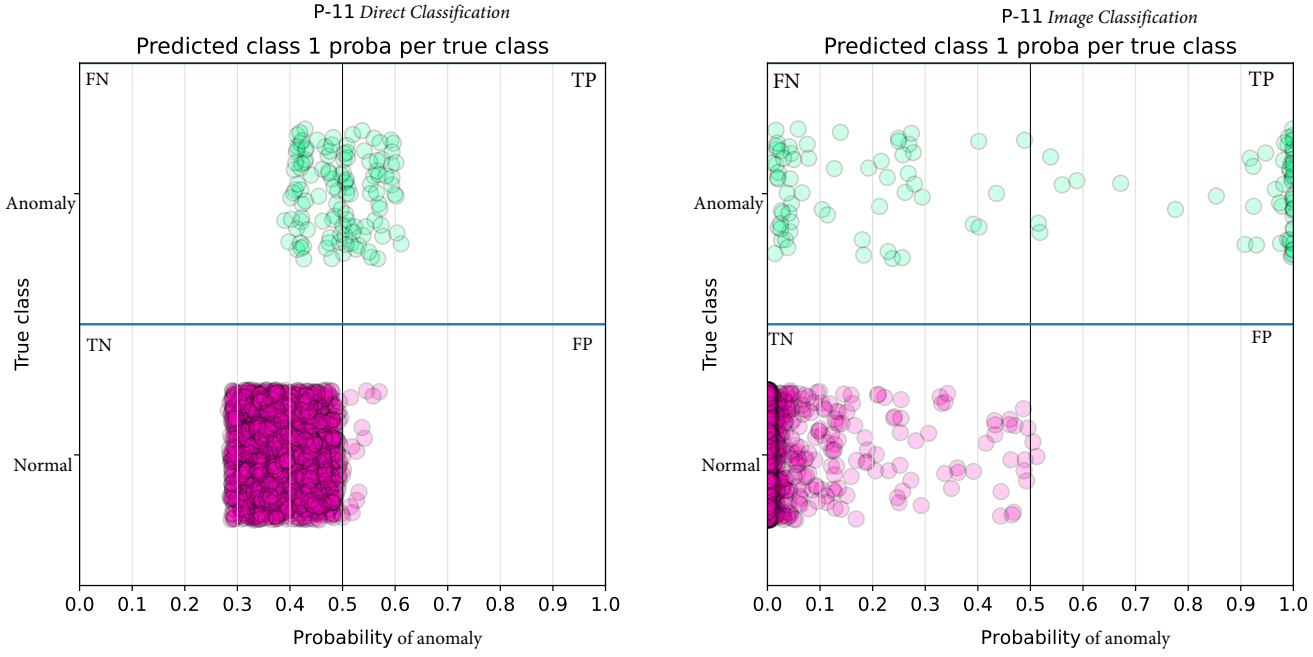
The *per-anomaly F1 score* was higher in *direct classification* than in *forecasting & threshold*. However, in *direct classification* the number of FP, i.e. false alarms, increased by a factor of 20 compared to *forecasting & threshold* [2] in particular state that avoiding FP is important for spacecraft anomaly detection. Moreover, [3] agree that FP are undesirable to an operational mission. Given the relatively high proportion of FP to TP in both *direct classification* and *image classification*, we must consider both approaches are promising but not yet suitable for spacecraft anomaly detection without further work.

TABLE II: *Forecasting & threshold vs. direct classification*

	Forecasting & Threshold				Direct classification			
Channel	**TP**	**FP**	**FN**	**F1 %**	**TP**	**FP**	**FN**	**F1 %**
SMAP								
E-1	1	0	0	100.0	1	0	0	100.0
E-12	1	0	0	100.0	0	2	1	100.0
E-11	1	0	0	100.0	1	0	0	100.0
E-10	1	0	0	100.0	1	0	0	100.0
E-13	0	0	1	0.0	1	6	0	14.3
G-7	1	0	0	100.0	1	0	0	100.0
P-1	0	0	1	0.0	1	3	0	25.0
P-4	0	0	1	0.0	1	0	0	100.0
T-1	0	1	1	0.0	1	0	0	100.0
T-3	1	0	0	100.0	1	0	0	100.0
Overall	6	**1**	4	**70.6**	**9**	11	**1**	60.0
MSL								
C-1	0	0	1	0.0	1	0	0	100.0
C-2	0	0	1	0.0	1	6	0	14.3
F-7	0	0	1	0.0	1	3	0	25.0
P-11	0	0	1	0.0	1	1	0	50.0
T-13	0	0	1	0.0	1	3	0	25.0
T-9	0	0	1	0.0	1	0	0	100.0
Overall	0	**0**	6	0.0	**6**	13	**0**	**48.0**
Total	6	**1**	10	52.2	**15**	24	**1**	**54.5**

TABLE III: *Image classification*

	Image classif. xResNet34				Image classif. xResNet34$_{\text{(fine-tuned)}}$			
Channel	**TP**	**FP**	**FN**	**F1 %**	**TP**	**FP**	**FN**	**F1 %**
SMAP								
E-1	0	0	1	0.0	0	0	1	0.0
E-12	0	0	1	0.0	0	0	1	0.0
E-11	0	0	1	0.0	0	0	1	0.0
E-10	0	0	1	0.0	0	0	1	0.0
E-13	1	15	0	11.8	0	0	1	0.0
G-7	1	0	0	100.0	1	0	0	100.0
P-1	1	16	0	11.1	1	16	0	11.1
P-4	1	0	0	100.0	1	0	0	100.0
T-1	1	36	0	5.26	1	36	0	5.26
T-3	0	0	1	0.0	1	0	0	100.0
Overall	**5**	67	**5**	12.2	**5**	**52**	**5**	**14.9**
MSL								
C-1	1	3	0	40.0	1	3	0	40.0
C-2	1	7	0	22.2	1	7	0	22.2
F-7	1	5	0	28.6	1	5	0	28.6
P-11	1	3	0	40.0	1	0	0	100.0
T-13	1	0	0	100.0	1	0	0	100.0
T-9	1	1	0	66.7	1	1	0	66.7
Overall	**6**	19	**0**	38.7	**6**	**16**	**0**	**42.9**
Total	11	86	**5**	19.5	**13**	**68**	**5**	**26.3**



(a) Probability of Anomaly, *Direct classification*, P-11

(b) Probability of Anomaly, *Image classification*, P-11

Fig. 6: Image classifier probability plots

Further analysing the telemetry channels shows that G-7—the only "spiky" telemetry channel (usually flat with occasional peaks) in the SMAP/MSL dataset—performs consistently well in all our 3 analysed approaches. This suggests that some feature or behavioural trait of spiky data makes anomaly detection more effective for deep learning. This is perhaps counter-intuitive, as one would expect *anomalous* signals to be masked by the spiky signals. Alternatively, it may be the case that the contextual information is particularly good at discriminating the relevant data points against the spiky normal signal. This shows the potential to solve difficult cases where humans may find it difficult to differentiate *anomalous* from *normal* data.

B. Understanding image classification performance

Whilst overall the *image classification* results were disappointing within the details of the model outputs, there are some interesting results which suggest there is potential in the approach. The *per-data point* results illustrated in Fig. 6 suggests there is scope for further improvement in *image classification*, at least for some types of data. Each sub-figure represents a single model and telemetry channel pair. Individual data points are plotted on the y-axis according to the true class, i.e. the actual or correct class label: green for true *anomalous* and purple for true *normal*; data points are spread vertically *within* the classes, so as to make the points more visible. The model's probability of any point being *anomalous* is plotted on the x-axis. The plots can be read as confusion matrices. The optimal case would be if all green points, indicating an anomaly, were on the right upper corner and all purple points, indicating the normal case, were on the left lower corner. This visualisation allows us to explain the behaviour of the classifiers by visualising how each data point was classified.

The *direct classification*/P-11 plot (Fig. 6a) shows that the two classes are barely separated (poor differentiation) and FPs are present. In contrast, the *image classification*/xResNet34$_{\text{(fine-tuned)}}$/P-11 plot (Fig. 6b) has good differentiation and demonstrates that there are no FP. This case is preferred for spacecraft operators [2], where FP increase workload and decrease trust in the anomaly detection system. The probability of the TP values is generally high, denoting a high confidence in the results. In fact, *image classification*/P-11 outperforms both *direct classification* and *forecasting & threshold*/P-11. A similar trend is observed for C-2 and G-7 (not plotted).

C. Discussion of limitations and challenges encountered

Our work has been limited primarily by the dataset chosen, the SMAP/MSL dataset. Despite it being the best dataset available as described in Section II-A, there are some fundamental difficulties associated with its use [19]. The anonymisation of the data has removed information as to the underlying nature of the data which precludes the use of domain knowledge to improve the model performance. The context data is provided as one-hot encoding of system-relevant commands. But this almost certainly leads to loss of useful information, such as any parameters associated to those commands and cross-coupling effects between telemetry channels. In a real spacecraft, where data is closely correlated [73], such context information is critical to understanding what may be an expected event versus a genuine anomaly.

Due to these issues with the SMAP/MSL dataset, we were obliged to follow the *one-model-per-channel* approach of [2]. In future applications, we would advocate to combine many related telemetry channels into a single model such as per spacecraft subsystem or per data type, as demonstrated in [13]. In this way the issue of "scalability" (Section V-A) may be reasonably addressed.

In the SMAP/MSL dataset, we have very few anomaly cases from which the deep learning system may learn, typically one anomaly in the training set and one in the test set, representing a few hundred *anomalous* data points compared to thousands *normal*. This leads to an extremely imbalanced dataset, which despite trying specialised loss functions [74] and weighting [75], resulted in overfitted models, indicated by the high number of FPs.

D. Computing time for training and inference

Model training times are a critical metric for spacecraft anomaly detection as it places an upper limit on how frequently the model could be re-trained. Ideally such a model would be retrained regularly to account for spacecraft aging and seasonal effects. Inference time is also important because it limits the reaction time of the spacecraft operators.

The training time of the *forecasting & threshold* model in [13] was around 3 hours. The *direct classification* model training time only was approximately 7 minutes. The *image classification* model training time, which includes the GAF transformation, was over 12 hours. This shows that an operator can retrain the *direct classification* much more frequently. In all cases the inference time was negligible, in the order of a few tens of seconds. This demonstrates that the inference time has no impact on the decision which model to chose. These times are based on the computational resources described in Section III-D.

VI. Conclusion

In this section, we will summarise our work and describe possible future steps.

A. Summary of research

Our experiments show that *direct classification* can perform better than the classic *forecasting & threshold* methods which are the backbone of the current approaches for spacecraft anomaly detection [1]–[4]. There is a clear difference between the performance of *direct classification* between SMAP and MSL, with SMAP performing better (60% vs. 48%). The *direct classification* of the MSL data significantly outperformed *forecasting & threshold* finding anomalies not discovered by the latter, as illustrated in Table II. The results overall exceed the state-of-the-art performance in recent studies [76]

and represent a new avenue for further exploration in future studies.

Image classification did not manage to meet the same level of performance as *direct classification*. Regardless, for some channels the results show better class separation than for *direct classification* (P-11, shown in Fig. 6). As spacecraft operators are used to visually inspecting plots to identify anomalies [1], it should not be a surprise that computer vision-based *image classification* can be successful, using image transforms to visualise the time series data. Challenges remain to properly understand the mechanics behind the transforms and optimise them for deep learning systems. There are two main drawbacks to *image classification*: slow speed of the transformations and a high number of FNs.

B. Future work

One possible avenue of investigation which may bear fruit in the case of spacecraft anomaly detection is one-class classification, which does not rely on seeing examples of both *anomalous* and *normal* data. Since anomalies appear rarely in spacecraft telemetry [16], [43], and *well labelled* anomalies are rarer still, a one-class classifier based only on normal data may yield better results than training a binary classifier on very few examples. "*A one-class classifier aims at capturing characteristics of training instances, in order to be able to distinguish between them and potential outliers to appear*" [77]. On the other hand, "*the advantages of one-class classifiers come at a price of discarding all of the available information about the majority class*" [77], namely the context in which the normal data occurs. Further research in this direction would be worthwhile.

A popular method to deal with severely imbalanced data in other fields is the use of oversampling. Oversampling approaches for time series data is an active area of research, such as [78], [79], and [80]. These attempt to create new instances of the minority classes whilst following the distribution of the original signal, such that the deep learning system has more to learn from. Unfortunately, at the time of our experiments the implementation of the oversampling systems has not been publicly released, so we were unable to include it in this study. None of the literature applies these oversampling techniques to spacecraft telemetry data however, so this would be an interesting and useful direction for future studies.

The use of a pre-trained (fine-tuned) xResNet34 model showed that in some cases a computer vision model can be effective for spacecraft anomaly detection, but not in all cases. Future work should be undertaken to establish the types and behaviour of data benefiting a *image classification* approach.

In [13] we demonstrated a novel technique to apply an unsupervised clustering algorithm on the telemetry channel data to determine a set of "data types", based on the shape of the signal. The idea behind this was to apply different shape-specific models to tailor the learning to the shape rather than using a single model for all. This greatly improved the results by allowing the use of an ensemble approach, applying the best performing deep learning architecture for that data type. It was not possible to apply the same technique with *direct classification* (or *image classification*) due to the reduced dataset leading to too few examples of most identified data shapes types. However, applying a clustering approach with our classification approaches could be investigated further with a suitable dataset.

More work would be needed to explore different image transforms for *image classification*, especially on the size of the transformed images. We took the common 224x224 image size popular in the literature [70]. But whether this is optimal for the selected window size is unclear. Ultimately, as a new approach not previously applied to the domain of spacecraft anomaly detection, the results are encouraging and show the utility of the system in principle for some types of data.

REFERENCES

[1] J. Heras and A. Donati, "Enhanced Telemetry Monitoring with Novelty Detection," *AI Magazine*, vol. 35, no. 4, pp. 37–46, 2014.

[2] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding," 2018.

[3] B. Pilastre, L. Boussouf, S. D'Escrivan, and J.-Y. Tourneret, "Anomaly Detection in Mixed Telemetry Data Using a Sparse Representation and Dictionary Learning," *Signal Processing*, vol. 168, p. 107320, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165168419303731

[4] S. Baireddy, S. R. Desai, J. L. Mathieson, R. H. Foster, M. W. Chan, M. L. Comer, and E. J. Delp, "Spacecraft Time-Series Anomaly Detection Using Transfer Learning," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021, pp. 1951–1960.

[5] S. Baireddy, S. R. Desai, R. H. Foster, M. W. Chan, M. L. Comer, and E. J. Delp, "Spacecraft Time-Series Online Anomaly Detection Using Deep Learning," in *2023 IEEE Aerospace Conference*. IEEE, 2023, pp. 1–9.

[6] Z. Wang and T. Oates, "Spatially Encoding Temporal Correlations to Classify Temporal Data Using Convolutional Neural Networks," 2015.

[7] L. Liu, L. Tian, Z. Kang, and T. Wan, "Spacecraft Anomaly Detection with Attention Temporal Convolution Network," 2023.

[8] S. Guan, B. Zhao, Z. Dong, M. Gao, and Z. He, "GTAD: Graph and Temporal Neural Network for Multivariate Time Series Anomaly Detection," *Entropy*, vol. 24, no. 6, p. 759, 2022. [Online]. Available: https://www.mdpi.com/1099-4300/24/6/759

[9] M. Rodríguez, D. P. Tobón, and D. Múnera, "Anomaly Classification in Industrial Internet of Things: A Review," *Intelligent Systems with Applications*, vol. 18, p. 200232, May 2023. [Online]. Available: http://dx.doi.org/10.1016/j.iswa.2023.200232

[10] F. Shahzad, A. Mannan, A. R. Javed, A. S. Almadhor, T. Baker, and D. Al-Jumeily OBE, "Cloud-based Multiclass Anomaly Detection and Categorization Using Ensemble Learning," *Journal of Cloud Computing*, vol. 11, no. 1, Nov. 2022. [Online]. Available: http://dx.doi.org/10.1186/s13677-022-00329-y

[11] G. Liu, Y. Niu, W. Zhao, Y. Duan, and J. Shu, "Data Anomaly Detection for Structural Health Monitoring Using a Combination Network of GANomaly and CNN," *Smart Struct. Syst*, vol. 29, no. 1, pp. 53–62, 2022. [Online]. Available: https://doi.org/10.12989/sss.2022.29.1.053

[12] R. Sakurai and T. Yairi, "Proposal of a Time Series Anomaly Detection Method Using Image Encoding Techniques," *PHM Society Asia-Pacific Conference*, vol. 4, no. 1, Sep. 2023. [Online]. Available: http://dx.doi.org/10.36001/phmap.2023.v4i1.3760

[13] D. Lakey and T. Schlippe, "A Comparison of Deep Learning Architectures for Spacecraft Anomaly Detection," in *Proceedings of the 2024 IEEE Conference on Aerospace*, 2024. [Online]. Available: https://doi.org/10.1109/AERO58975.2024.10521015

[14] T. Yairi, T. Oda, Y. Nakajima, N. Miura, and N. Takata, "Evaluation Testing of Learning-based Telemetry Monitoring and Anomaly Detection System in SDS-4 Operation," in *Proceedings of the International Symposium on Artificial Intelligence, Robotics and*

Automation in Space (i-SAIRAS), 2014, [Accessed 28-10-2023]. [Online]. Available: https://api.semanticscholar.org/CorpusID:36521312

[15] P. Fortescue, G. Swinerd, and J. Stark, *Spacecraft Systems Engineering*, 4th ed. Nashville, TN: John Wiley & Sons, 2011.

[16] R. R. Lutz and I. C. Mikulski, "Empirical Analysis of Safety-critical Anomalies During Operations," *IEEE Transactions on Software Engineering*, vol. 30, no. 3, pp. 172–180, 2004.

[17] P. Benecki, S. Piechaczek, D. Kostrzewa, and J. Nalepa, "Detecting Anomalies in Spacecraft Telemetry Using Evolutionary Thresholding and LSTMs," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2021, pp. 143–144.

[18] T. H. H. Lu, *Semi-Supervised Deep Learning for Spacecraft Anomaly Detection*. McGill University (Canada), 2022.

[19] R. Wu and E. Keogh, "Current Time Series Anomaly Detection Benchmarks are Flawed and are Creating the Illusion of Progress," *IEEE Transactions on Knowledge and Data Engineering*, p. 1, 2021.

[20] P. O'Neill, D. Entekhabi, E. Njoku, and K. Kellogg, "The NASA Soil Moisture Active Passive (SMAP) Mission: Overview," in *2010 IEEE International Geoscience and Remote Sensing Symposium*, 2010, pp. 3236–3239.

[21] A. R. Vasavada, "Mission Overview and Scientific Contributions from the Mars Science Laboratory Curiosity Rover After Eight Years of Surface Operations," *Space Science Reviews*, vol. 218, no. 3, Apr. 2022. [Online]. Available: https://doi.org/10.1007/s11214-022-00882-7

[22] A. Ihler, J. Hutchins, and P. Smyth, "Adaptive Event Detection with Time-varying Poisson Processes," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, August 2006. [Online]. Available: https://doi.org/10.1145/1150402.1150428

[23] M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed, "DeepAnT: A Deep Learning Approach for Unsupervised Anomaly Detection in Time Series," *IEEE Access*, vol. 7, pp. 1991–2005, 2019. [Online]. Available: https://doi.org/10.1109/access.2018.2886457

[24] P. Wenig, S. Schmidl, and T. Papenbrock, "TimeEval: A Benchmarking Toolkit for Time Series Anomaly Detection Algorithms," vol. 15, no. 12, pp. 3678–3681, 2022.

[25] S. Schmidl, P. Wenig, and T. Papenbrock, "Anomaly Detection in Time Series: A Comprehensive Evaluation," *Proc. VLDB Endow.*, vol. 15, no. 9, p. 1779–1797, may 2022. [Online]. Available: https://doi.org/10.14778/3538598.3538602

[26] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and practice*. [Australia]: OTexts, 2018. [Online]. Available: https://otexts.com/fpp3/

[27] Z. Wang, W. Yan, and T. Oates, "Time Series Classification from Scratch with Deep Neural Networks: A Strong Baseline," 2016.

[28] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep Learning for Time Series Classification: A Review," *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, Mar. 2019. [Online]. Available: https://doi.org/10.1007/s10618-019-00619-1

[29] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "LSTM Fully Convolutional Networks for Time Series Classification," *IEEE Access*, vol. 6, pp. 1662–1669, 2018. [Online]. Available: https://doi.org/10.1109/access.2017.2779939

[30] F. Harrell, "Classification vs. Prediction," Jan 2017. [Online]. Available: https://www.fharrell.com/post/classification/

[31] E. Eldele, M. Ragab, Z. Chen, M. Wu, C. K. Kwoh, X. Li, and C. Guan, "Time-Series Representation Learning via Temporal and Contextual Contrasting," 2021. [Online]. Available: https://arxiv.org/abs/2106.14112

[32] B. H. D. Koh, C. L. P. Lim, H. Rahimi, W. L. Woo, and B. Gao, "Deep Temporal Convolution Network for Time Series Classification," *Sensors*, vol. 21, no. 2, p. 603, Jan. 2021. [Online]. Available: http://dx.doi.org/10.3390/s21020603

[33] B. Altaf, L. Yu, and X. Zhang, "Spatio-Temporal Attention Based Recurrent Neural Network for Next Location Prediction," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, Dec. 2018. [Online]. Available: http://dx.doi.org/10.1109/BigData.2018.8622218

[34] A. Kulshrestha, L. Chang, and A. Stein, "Use of LSTM for Sinkhole-Related Anomaly Detection and Classification of InSAR Deformation Time Series," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 4559–4570, 2022. [Online]. Available: https://doi.org/10.1109/jstars.2022.3180994

[35] N. Sharma, V. Jain, and A. Mishra, "An Analysis Of Convolutional Neural Networks For Image Classification," *Procedia Computer Science*, vol. 132, pp. 377–384, 2018. [Online]. Available: https://doi.org/10.1016/j.procs.2018.05.198

[36] B. Staar, M. Lütjen, and M. Freitag, "Anomaly Detection with Convolutional Neural Networks for Industrial Surface Inspection," *Procedia CIRP*, vol. 79, pp. 484–489, 2019. [Online]. Available: https://doi.org/10.1016/j.procir.2019.02.123

[37] Z. Wang and T. Oates, "Encoding Time Series as Images for Visual Inspection and Classification Using Tiled Convolutional Neural Networks," 2014, [Accessed 27-10-2023]. [Online]. Available: https://api.semanticscholar.org/CorpusID:16409971

[38] W. Jiang, D. Zhang, L. Ling, and R. Lin, "Time Series Classification Based on Image Transformation Using Feature Fusion Strategy," *Neural Processing Letters*, vol. 54, no. 5, p. 3727–3748, Mar. 2022. [Online]. Available: http://dx.doi.org/10.1007/s11063-022-10783-z

[39] V. Sreeram and P. Agathoklis, "On the Properties of Gram Matrix," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 41, no. 3, pp. 234–237, Mar. 1994. [Online]. Available: https://doi.org/10.1109/81.273922

[40] L. d. Vitry, "Encoding time series as images," *Medium*, Oct 2018, [Accessed 30-10-2023]. [Online]. Available: https://medium.com/analytics-vidhya/encoding-time-series-as-images-b043becbdbf3

[41] S. Barra, S. M. Carta, A. Corriga, A. S. Podda, and D. R. Recupero, "Deep Learning and Time Series-to-Image Encoding for Financial Forecasting," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 3, pp. 683–692, 2020.

[42] C. S. Sastry and S. Oore, "Detecting Out-of-Distribution Examples with In-distribution Examples and Gram Matrices," 2019. [Online]. Available: https://arxiv.org/abs/1912.12510

[43] N. Iucci, A. E. Levitin, A. V. Belov, E. A. Eroshenko, N. G. Ptitsyna, G. Villoresi, G. V. Chizhenkov, L. I. Dorman, L. I. Gromova, M. Parisi, M. I. Tyasto, and V. G. Yanke, "Space Weather Conditions and Spacecraft Anomalies in Different Orbits," *Space Weather*, vol. 3, no. 1, Jan. 2005. [Online]. Available: https://doi.org/10.1029/2003sw000056

[44] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, "The Great Time Series Classification Bake Off: A Review and Experimental Evaluation of Recent Algorithmic Advances," *Data Mining and Knowledge Discovery*, vol. 31, no. 3, pp. 606–660, Nov. 2016. [Online]. Available: https://doi.org/10.1007/s10618-016-0483-9

[45] H. A. Dau, A. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. Keogh, "The UCR Time Series Archive," 2018. [Online]. Available: https://arxiv.org/abs/1810.07758

[46] L. Kulanuwat, C. Chantrapornchai, M. Maleewong, P. Wongchaisuwat, S. Wimala, K. Sarinnapakorn, and S. Boonya-aroonnet, "Anomaly Detection Using a Sliding Window Technique and Data Imputation with Machine Learning for Hydrological Time Series," *Water*, vol. 13, no. 13, p. 1862, Jul. 2021. [Online]. Available: http://dx.doi.org/10.3390/w13131862

[47] W.-S. Hwang, J.-H. Yun, J. Kim, and B. G. Min, "Do You Know Existing Accuracy Metrics Overrate Time-series Anomaly Detections?" in *The 37th Annual ACM Symposium on Applied Computing*, J. Hong, M. Bures, J. W. Park, and T. Cerny, Eds. New York, NY, United States: Association for Computing Machinery, 2022, pp. 403–412.

[48] A. Huet, J. M. Navarro, and D. Rossi, "Local Evaluation of Time Series Anomaly Detection Algorithms," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, August 2022. [Online]. Available: https://doi.org/10.1145/3534678.3539339

[49] C. Goutte and E. Gaussier, "A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation," in *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2005, pp. 345–359. [Online]. Available: https://doi.org/10.1007/978-3-540-31865-1_25

[50] D. Berrar and P. Flach, "Caveats and Pitfalls of ROC Analysis in Clinical Microarray Research (and how to Avoid Them)," *Briefings in Bioinformatics*, vol. 13, no. 1, pp. 83–97, Mar. 2011. [Online]. Available: https://doi.org/10.1093/bib/bbr008

[51] E. Rahimian, S. Zabihi, S. F. Atashzar, A. Asif, and A. Mohammadi, "XceptionTime: A Novel Deep Architecture based on Depthwise Separable Convolutions for Hand Gesture Classification," 2019. [Online]. Available: https://arxiv.org/abs/1911.03803

[52] I. Oguiza, "tsai - A State-of-the-Art Deep Learning Library for Time Series and Sequential Data," Github, 2022, [Accessed 28-10-2023]. [Online]. Available: https://github.com/timeseriesAI/tsai

[53] L. Bickmann, L. Plagwitz, and J. Varghese, "Post Hoc Sample Size Estimation for Deep Learning Architectures for ECG-Classification,"

Stud. Health Technol. Inform, vol. 302, pp. 182–186, 2023. [Online]. Available: https://ebooks.iospress.nl/pdf/doi/10.3233/SHTI230099

[54] A. Nazir, R. Mitra, H. Sulieman, and F. Kamalov, "Suspicious Behavior Detection with Temporal Feature Extraction and Time-Series Classification for Shoplifting Crime Prevention," *Sensors*, vol. 23, no. 13, p. 5811, Jun. 2023. [Online]. Available: http://dx.doi.org/10.3390/s23135811

[55] E. Bisong, "Google Colaboratory," in *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Apress, 2019, pp. 59–64. [Online]. Available: https://doi.org/10.1007/978-1-4842-4470-8_7

[56] Google Colabratory, "GPU Architecture," 2020, [Accessed 05-11-2023]. [Online]. Available: https://colab.research.google.com/github/d2l-ai/d2l-tvm-colab/blob/master/chapter_gpu_schedules/arch.ipynb

[57] J. Faouzi and H. Janati, "pyts: A Python Package for Time Series Classification," *Journal of Machine Learning Research*, vol. 21, no. 46, pp. 1–6, 2020, [Accessed 31-10-2023]. [Online]. Available: http://jmlr.org/papers/v21/19-763.html

[58] J.-P. Eckmann, S. O. Kamphorst, and D. Ruelle, "Recurrence Plots of Dynamical Systems," *Europhysics Letters (EPL)*, vol. 4, no. 9, pp. 973–977, Nov. 1987. [Online]. Available: https://doi.org/10.1209/0295-5075/4/9/004

[59] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2015.

[60] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of Tricks for Image Classification with Convolutional Neural Networks," 2018. [Online]. Available: https://arxiv.org/abs/1812.01187

[61] M. Gao, D. Qi, H. Mu, and J. Chen, "A Transfer Residual Neural Network Based on ResNet-34 for Detection of Wood Knot Defects," *Forests*, vol. 12, no. 2, p. 212, Feb. 2021. [Online]. Available: https://doi.org/10.3390/f12020212

[62] Q. Zhuang, S. Gan, and L. Zhang, "Human-Computer Interaction Based Health Diagnostics Using ResNet34 for Tongue Image Classification," *Computer Methods and Programs in Biomedicine*, vol. 226, p. 107096, Nov. 2022. [Online]. Available: https://doi.org/10.1016/j.cmpb.2022.107096

[63] L. Gao, X. Zhang, T. Yang, B. Wang, and J. Li, "The Application of ResNet-34 Model Integrating Transfer Learning in the Recognition and Classification of Overseas Chinese Frescoes," *Electronics*, vol. 12, no. 17, p. 3677, Aug. 2023. [Online]. Available: https://doi.org/10.3390/electronics12173677

[64] Pytorch, "Models and Pre-trained Weights; Torchvision 0.17 Documentation - pytorch.org," https://pytorch.org/vision/stable/models.html, [Accessed 06-02-2024].

[65] S. Shahinfar, P. Meek, and G. Falzon, ""How Many Images Do I need?" Understanding How Sample Size per Class Affects Deep Learning Model Performance Metrics for Balanced Designs in Autonomous Wildlife Monitoring," *Ecological Informatics*, vol. 57, p. 101085, May 2020. [Online]. Available: http://dx.doi.org/10.1016/j.ecoinf.2020.101085

[66] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 2014. [Online]. Available: https://arxiv.org/abs/1409.1556

[67] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," 2017. [Online]. Available: https://arxiv.org/abs/1704.04861

[68] Y. Zhang, X. Gao, Q. Duan, J. Leng, X. Pu, and X. Gao, "Contextual Learning in Fourier Complex Field for VHR Remote Sensing Images," 2022. [Online]. Available: https://arxiv.org/abs/2210.15972

[69] M. L. Richter, W. Byttner, U. Krumnack, A. Wiedenroth, L. Schallner, and J. Shenk, *(Input) Size Matters for CNN Classifiers*. Springer International Publishing, 2021, p. 133–144. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-86340-1_11

[70] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[71] R. F. Woolson, "Wilcoxon Signed-Rank Test," p. 1–3, Sep. 2008. [Online]. Available: http://dx.doi.org/10.1002/9780471462422.eoct979

[72] W. J. Dixon and A. M. Mood, "The Statistical Sign Test," *Journal of the American Statistical Association*, vol. 41, no. 236, p. 557–566, Dec. 1946. [Online]. Available: http://dx.doi.org/10.1080/01621459.1946.10501898

[73] S. K. Ibrahim, A. Ahmed, M. A. E. Zeidan, and I. E. Ziedan, "Machine Learning Methods for Spacecraft Telemetry Mining," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 4, p. 1816–1827, August 2019. [Online]. Available: http://dx.doi.org/10.1109/TAES.2018.2876586

[74] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," 2017. [Online]. Available: https://arxiv.org/abs/1708.02002

[75] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-Balanced Loss Based on Effective Number of Samples," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2019. [Online]. Available: https://doi.org/10.1109/cvpr.2019.00949

[76] C. Wang, Q. Liu, H. Zhou, T. Wu, H. Liu, J. Huang, Y. Zhuo, Z. Li, and K. Li, "Anomaly Prediction of CT Equipment Based on IoMT Data," *BMC Medical Informatics and Decision Making*, vol. 23, no. 1, Aug. 2023. [Online]. Available: https://doi.org/10.1186/s12911-023-02267-4

[77] A. Fernandez, S. Garcia, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning from Imbalanced Data Sets*, 1st ed. Cham, Switzerland: Springer International Publishing, Nov. 2018.

[78] P. Liu, X. Guo, R. Wang, P. Chen, T. Wo, and X. Liu, "CSMOTE: Contrastive Synthetic Minority Oversampling for Imbalanced Time Series Classification," in *Neural Information Processing*, ser. Communications in Computer and Information Science, T. Mantoro, M. Lee, M. A. Ayu, K. W. Wong, and A. N. Hidayanto, Eds. Cham: Springer International Publishing, 2021, vol. 1516, pp. 447–455.

[79] Pu Zhao, Chuan Luo, Bo Qiao, Lu Wang, Saravan Rajmohan, Qingwei Lin, and Dongmei Zhang, *T-SMOTE: Temporal-oriented Synthetic Minority Oversampling Technique for Imbalanced Time Series Classification*, ser. Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22), 2022.

[80] A. Baumgartner, S. Molani, Q. Wei, and J. Hadlock, "Imputing Missing Observations with Time Sliced Synthetic Minority Oversampling Technique," 2022. [Online]. Available: https://arxiv.org/abs/2201.05634