

# Deep Learning-Based Anomaly Detection in Spacecraft Telemetry on Edge Devices

Christopher Goetze 

IU International University  
of Applied Sciences  
Germany

christopher.goetze@iu-study.org  
ORCID:0009-0001-2465-5701

Tim Schlippe 

IU International University  
of Applied Sciences  
Germany

tim.schlippe@iu.org  
ORCID:0000-0002-9462-8610

Daniel Lakey 

IU International University  
of Applied Sciences  
Germany

daniel.lakey@iu-study.org  
ORCID:0000-0002-8198-7892

**Abstract**—Spacecraft anomaly detection is critical for mission safety, yet deploying sophisticated models on-board presents significant challenges due to hardware constraints. This paper investigates three approaches for spacecraft telemetry anomaly detection—*forecasting & threshold*, *direct classification*, and *image classification*—and optimizes them for edge deployment using multi-objective neural architecture optimization on the European Space Agency Anomaly Dataset. Our baseline experiments demonstrate that *forecasting & threshold* achieves superior detection performance (92.7% Corrected Event-wise  $F_{0.5}$ -score ( $CEF_{0.5}$ )) [1] compared to alternatives. Through *Pareto-optimal* architecture optimization, we dramatically reduced computational requirements while maintaining capabilities—the optimized *forecasting & threshold* model preserved 88.8%  $CEF_{0.5}$  while reducing RAM usage by 97.1% to just 59 KB and operations by 99.4%. Analysis of deployment viability shows our *optimized* models require just 0.36-6.25% of CubeSat RAM, making on-board anomaly detection practical even on highly constrained hardware. This research demonstrates that sophisticated anomaly detection capabilities can be successfully deployed within spacecraft edge computing constraints, providing near-instantaneous detection without exceeding hardware limitations or compromising mission safety.

**Index Terms**—anomaly detection, spacecraft telemetry, time series classification, image classification, edge devices

## I. INTRODUCTION

Space missions rely critically on the health and operational status of spacecraft systems, making anomaly detection in telemetry data essential for mission success and longevity [2], [3]. Modern spacecraft generate vast amounts of telemetry data—thousands of parameters sampled at frequencies ranging from seconds to minutes—creating an overwhelming volume of information that cannot be manually monitored [4]. This challenge is compounded by the extreme rarity of actual anomalies, with a comprehensive study of seven spacecraft over a decade identifying fewer than 200 critical anomalies [5].

Traditional ground-based anomaly detection approaches process telemetry data after transmission to Earth, introducing potentially critical delays ranging from seconds for low-Earth orbit missions to hours for deep space missions [3]. These delays can be substantially longer if there are significant periods when the spacecraft is outside of ground-station visibility [6], [7], potentially compromising mission safety when immediate intervention is required [8]. Moreover, bandwidth limitations

often necessitate downsampling or selective transmission of telemetry data, potentially obscuring subtle anomalies [8].

On-board anomaly detection offers a compelling alternative by analyzing telemetry data directly on the spacecraft, enabling near-instantaneous detection and response to critical events. This approach minimizes communication delays, conserves limited downlink bandwidth, and allows spacecraft to implement autonomous recovery actions when Earth communication is unavailable [2], [7]. However, deploying sophisticated anomaly detection systems on spacecraft presents significant challenges due to the severe computational constraints of space-grade hardware [3].

Space-qualified computing systems typically offer just a fraction of the processing power, memory, and storage available in terrestrial environments [8]. Additionally, these systems must operate reliably within tight power budgets and withstand harsh radiation environments [3]. These constraints preclude the direct deployment of state-of-the-art anomaly detection models developed for ground-based applications, necessitating specialized approaches for edge implementation.

In this paper, we investigate three distinct deep learning-based approaches for spacecraft telemetry anomaly detection:

- *forecasting & threshold*, which predicts normal behavior and identifies deviations;
- *direct classification*, which directly labels telemetry segments as normal or anomalous; and
- *image classification*, which transforms time series data into images for visual pattern recognition.

We conduct comprehensive evaluations using the European Space Agency Benchmark for Anomaly Detection in Satellite Telemetry (ESA-ADB) [1], containing real operational data from three distinct missions with expert-annotated anomalies.

Beyond comparing model performance, we specifically address the critical challenge of deploying these solutions on space-grade edge devices. We employ neural architecture optimization techniques to systematically reduce model size and computational requirements while preserving detection accuracy. This optimization process leverages multi-objective frameworks that balance performance against hardware constraints, enabling deployment on severely resource-constrained spacecraft computing systems.

Our research makes three primary contributions:

- a comprehensive comparison of deep learning approaches for spacecraft telemetry anomaly detection using real operational data;
- a systematic methodology for optimizing these models for deployment on space-grade edge devices; and
- empirical evidence demonstrating that sophisticated anomaly detection can be achieved within the stringent computational constraints of spacecraft hardware.

The remainder of this paper is organized as follows: Section II reviews related work in anomaly detection, spacecraft telemetry datasets, and neural architecture optimization. Section III details our experimental setup, including the models, dataset, and evaluation metrics. Section IV presents our experimental results and discusses the implications for edge deployment. Finally, Section V concludes the paper and outlines future research directions.

## II. RELATED WORK

This section examines current research in spacecraft telemetry anomaly detection, neural architecture optimization for resource-constrained environments, and available spacecraft telemetry datasets.

### A. Anomaly Detection in Spacecraft Telemetry

Anomaly detection in spacecraft telemetry is critical for mission safety and success [2], [4], [9]–[11]. Three primary approaches have emerged in this domain: *forecasting & threshold*, *direct classification*, and *image classification*. Recent surveys like [12] provide comprehensive overviews of deep learning techniques for time series anomaly detection across multiple domains, highlighting the increasing importance of these methods for critical systems monitoring.

1) *Forecasting & Threshold*: The *forecasting & threshold* approach predicts future values based on historical data and identifies anomalies when actual values deviate beyond a specified threshold. The model forecasts values based on learned patterns from previous timesteps, then compares these predictions with observations to quantify anomalousness. This method requires training on normal data to establish baseline behavior patterns. Various models have been adapted for this purpose, including statistical methods like AutoRegressive Integrated Moving Average (ARIMA) [13] and deep learning architectures such as Telemanom’s Long Short-Term Memories (LSTMs) [2] and DeepAnT’s Convolutional Neural Networks (CNNs) [14]. [15] demonstrated significant improvements in spacecraft anomaly detection by combining LSTM networks with multi-scale detection strategies, validating their approach on NASA benchmark data and Beidou Navigation Satellite hydrogen clock telemetry. Their work highlights how enhanced deep learning architectures can significantly improve operational reliability through early anomaly detection.

The effectiveness of this approach depends heavily on threshold selection. [2] proposed an “unsupervised and non-parametric anomaly thresholding approach” that dynamically learns appropriate error thresholds for each telemetry channel.

While over 158 time series anomaly detection models existed as of 2022 [16], many are designed for univariate data [17], limiting their applicability to the multivariate context of spacecraft telemetry.

2) *Direct Classification*: *Direct classification* approaches label time points or windows directly as normal or anomalous without intermediate forecasting steps. This method circumvents the challenges of complex threshold tuning and forecasting inaccuracies [18]. Deep learning architectures like CNNs [19]–[21] and RNNs [22], [23] have demonstrated promising results in capturing temporal dynamics for classification tasks. As detailed in the comprehensive survey by [12], these *direct classification* approaches have gained significant traction across multiple domains due to their ability to learn complex patterns directly from data without manual feature engineering. Recent research by [24] indicates that *direct classification* provides statistically significant improvements over traditional forecasting methods for spacecraft telemetry.

3) *Image Classification*: The *image classification* approach transforms time series data into images before applying classification algorithms. This leverages the exceptional performance of CNNs in image recognition [25], [26]. A prominent transformation technique is the Gramian Angular Field (GAF) [27]–[29], which encodes time series into polar coordinates and then into a two-dimensional representation using Gram matrices [30].

GAF transformations preserve temporal dependencies by encoding time into the geometry of the matrix [30]. This approach has been successfully applied in financial forecasting [31] and general anomaly detection [32]. While [24] found *image classification* less effective than *direct classification* for spacecraft telemetry, they suggest it offers promising directions for future research in this domain.

### B. Neural Architecture Search and Optimization

Deploying deep learning models for spacecraft anomaly detection on edge devices requires addressing severe hardware constraints [3], [8]. Unlike terrestrial systems that can be upgraded, spacecraft computing resources remain fixed throughout the mission lifetime, making efficient resource utilization critical [3].

CubeSat [33] and OPS-SAT [34], [35] are two prevalent on-board computing platforms utilized in contemporary spacecraft, each catering to different operational scopes and resource capabilities. CubeSats are compact, standardized satellites designed for cost-effective, modular deployment in space research missions. A typical CubeSat features modest computational resources, including a PIC32MZ M14K processor operating at speeds of up to 200 MHz, along with 16 MB of Random Access Memory (RAM) and 64 MB of flash memory [33]. In contrast, ESA’s OPS-SAT serves as an advanced, flexible experimentation platform equipped with more robust hardware, making it suitable for demanding computational tasks. Specifically, OPS-SAT incorporates an Altera Cyclone V System-on-Chip (SoC) featuring an ARM dual-core 800 MHz Cortex-A9 processor, complemented by

1 GB DDR3 RAM and an external 8 GB industrial-grade SD card storage [35]. Given these limited computational resources, particularly in the CubeSat environment, deploying an anomaly detection model necessitates careful optimization. Minimizing the model’s resource footprint ensures that sufficient processing power is available for critical spacecraft subsystems, thereby maintaining operational reliability.

A fundamental concept in balancing competing objectives like model accuracy and computational efficiency is the *Pareto front* approach to multi-objective optimization [36]. *Pareto-optimal* represents a set of non-dominated solutions where a solution  $A$  dominates solution  $B$  if  $A$  is at least as good as  $B$  in all objectives and strictly better in at least one. Solutions on this front cannot be improved in any objective without degrading performance in another, representing optimal trade-offs between conflicting goals [36]. This approach is particularly valuable for spacecraft applications where designers must balance detection accuracy against strict computational limitations, though it does not prescribe a single optimal solution but rather presents a set of efficient alternatives from which mission designers can select based on specific deployment constraints.

*Neural architecture search* and *neural architecture optimization* represent two approaches to this challenge. *Neural architecture search* focuses on discovering entirely new architectures, exemplified by MicroNAS [37] and  $\mu$ NAS [38]. While  $\mu$ NAS has shown impressive results—improving accuracy by up to 4.8% while reducing memory usage by 4–13×—it requires substantial computational resources (23 GPU days for CIFAR10) and generates entirely new, unproven architectures.

For spacecraft applications, *neural architecture optimization* often provides a more practical approach by optimizing proven architectures rather than creating new ones, which is particularly important where reliability is critical [39]. *Multi-objective optimization* frameworks like Optuna [40] are especially valuable for this task. Optuna offers intelligent sampling algorithms that simultaneously consider both model accuracy and computational constraints, making it ideal for spacecraft applications with strict resource limitations.

Effective visualization and profiling tools are essential components of the architecture optimization process. Two notable examples are the CNN Analyzer and the MLTK Model Profiler. [41] utilized their CNN Analyzer to provide layer-wise insights into how different scaling factors affect model size, peak memory usage, and inference time. This dashboard-based tool enabled them to systematically optimize model architecture scaling factors, resulting in a 20.5% reduction in model size while simultaneously increasing accuracy by 4.0%. Similarly, the MLTK Model Profiler [42] serves the same fundamental purpose of visualizing and analyzing model efficiency, but with features specifically designed for space-grade embedded systems. Both tools enable developers to identify optimization opportunities by providing detailed breakdowns of resource consumption, but the MLTK Model Profiler offers additional integration with spacecraft hardware development

workflows, making it particularly suitable for our application in spacecraft telemetry anomaly detection.

In our work, we combined Optuna’s optimization capabilities with the detailed insights provided by the MLTK Model Profiler to optimize our anomaly detection models for spacecraft edge devices. Utilizing the *Pareto-optimal* approach for multi-objective optimization [36], we systematically balanced model performance against computational constraints. This *neural architecture optimization* approach enabled us to develop sophisticated anomaly detection capabilities that operate within the strict limitations of on-board hardware, making it possible to deploy effective deep learning models for on-board analysis without exceeding the stringent memory and computational constraints of space-grade computing systems.

### C. Spacecraft Telemetry Data

Spacecraft telemetry data presents unique challenges for anomaly detection due to its high dimensionality and the rarity of actual anomalies. Modern spacecraft generate data from tens of thousands of telemetry channels [43], producing volumes that exceed human monitoring capabilities [4], [44]. Spacecraft are designed to be robust and fault-tolerant, with extensive testing to minimize anomalies [45], resulting in extremely imbalanced datasets where anomalous events are scarce—a study across seven spacecraft over more than a decade identified fewer than 200 critical anomalies [5].

A significant obstacle in spacecraft anomaly detection research has been the scarcity of publicly available datasets. The SMAP/MSL dataset from [2] has become a common benchmark, containing 82 multivariate telemetry channels from the Soil Moisture Active Passive spacecraft and Mars Science Laboratory with approximately 100 labeled anomalies. However, this dataset has recognized limitations, including anonymized channel identifiers and normalized values [24], [46].

More recently, the European Space Agency Benchmark for Anomaly Detection in Satellite Telemetry (ESA-ADB) has emerged as a valuable resource for researchers [1]. It represents the first large-scale, real-life satellite telemetry dataset with curated anomaly annotations from three European Space Agency missions. Developed through an 18-month collaboration between Airbus Defence and Space, KP Labs, and ESA’s European Space Operations Centre, this dataset aims to provide a common benchmark for researchers across academia, research institutes, space agencies, and industry. The ESA-ADB is part of the broader Artificial Intelligence for Automation (A<sup>2</sup>I) Roadmap initiative launched in 2021 to leverage AI for automating space operations. In this work, we utilize the ESA-ADB for our experiments, leveraging its comprehensive real-world telemetry data to develop and validate our edge-optimized anomaly detection models.

## III. EXPERIMENTAL SETUP

### A. Forecasting & Threshold Models

For our forecasting-based anomaly detection approach, we built upon our previous research [47] where we extended the

Telemanom framework originally developed by [2]. While the original implementation relied exclusively on LSTM-based forecasting with a non-parametric thresholding mechanism, our prior work comprehensively evaluated 13 different deep learning architectures as potential forecasting components within this framework.

Our experiments identified XceptionTimePlus as the optimal architecture for spacecraft telemetry forecasting [24]. This model, based on the XceptionTime architecture [48] from the `tsai` time series analysis framework [49], leverages convolutional neural networks with depthwise separable convolutions to capture temporal patterns efficiently. This CNN-based approach demonstrated superior performance compared to the original LSTM implementation across multiple telemetry channels.

In the current study, we evaluate this optimized *forecasting & threshold* system (XceptionTimePlus with non-parametric thresholding) on the ESA-ADB. This approach serves as our *baseline* for comparison against the *direct classification* and *image classification* techniques described in subsequent sections.

## B. Direct Classification

For our *direct classification* approach, we implemented the XceptionTimePlus architecture, a variant of the XceptionTime model [48] available in the `tsai` framework [49]. We configured the model using the optimal hyperparameters identified in our previous study on the SMAP/MSL dataset [47]. This choice is well-supported by existing literature, as XceptionTime has demonstrated superior performance in multiple non-spacecraft anomaly detection studies [50], [51] and shows excellent capability in handling multivariate time series data, making it particularly suitable for spacecraft telemetry analysis.

Our implementation leveraged the Python time series classification framework `tsai` [49], executed within Kaggle Notebooks<sup>1</sup> utilizing an Nvidia P100 GPU<sup>2</sup>. We developed a comprehensive classification pipeline that:

- Configures the appropriate loss function, optimizer, and class weights
- Instantiates the XceptionTimePlus model with optimal parameters
- Performs direct binary classification (normal/anomalous) on time windows (224 time points as in [24])
- Evaluates performance metrics including true positives, false positives, false negatives, and Corrected Event-wise  $F_{0.5}$ -score [1]

The same pipeline architecture, with an additional transformation stage, was later adapted for our *image classification* experiments, providing a consistent experimental framework across all our anomaly detection approaches.

## C. Image Classification

Image encoding of time series data represents an innovative approach to anomaly detection that leverages the power of convolutional neural networks by transforming temporal data into spatial representations [19], [52]. For our implementation, we utilized the Gramian Angular Field (GAF) transformation from the `pyts` framework [53] to convert telemetry time series into image-based representations suitable for classification.

We developed a custom PyTorch-accelerated transformation module implementation<sup>3</sup> that significantly improves upon the original `pyts`. Our implementation eliminates unnecessary data reshaping and CPU-GPU transfers, performs calculations directly in batched operations on the target device, and uses vectorized operations with Einstein Summation Notation for matrix calculations. This optimization achieved an 80× speed-up with GPU acceleration compared to the original implementation, critically enabling efficient image transformation for our edge deployment targets.

The standard GAF transformation is designed for univariate (single parameter) time series data, presenting a challenge for spacecraft telemetry which is inherently multivariate with multiple interconnected parameters. To address this limitation while preserving the critical contextual relationships between parameters, we implemented a multi-channel approach. This method generates separate GAF images for each parameter within a telemetry channel, creating a “stack” of images that collectively preserves the multivariate nature of the data.

We explored alternative multivariate image transformations such as the recurrence plot method [54] available in `pyts`, but our preliminary experiments yielded poor results with no anomalies detected, leading us to focus exclusively on the stacked GAF approach.

For the classification of these transformed images, we implemented two variants of xResNet34, a `tsai` adaptation of the ResNet34 architecture [55], [56] known for its exceptional performance in *image classification* tasks [57]–[59]. Unlike XceptionTimePlus used in our *direct classification* approach, ResNet34 is specifically designed to handle multi-dimensional image inputs, making it ideal for our transformed telemetry data.

We evaluated both an untrained xResNet34 model trained from scratch on our telemetry images and a pre-trained version (referred to as xResNet34<sub>(fine-tuned)</sub>) from the `PyTorch` collection [60] that we fine-tuned on our dataset. The fine-tuned approach was expected to provide better performance given the relatively limited number of training samples available for this task [61].

Our implementation standardized on  $224 \times 224$  pixel images for the GAF transformations, a resolution widely recognized in the literature as providing an optimal balance of detail and computational efficiency [62]–[65]. This resolution also aligned with the pre-training dimensions of the xResNet34<sub>(fine-tuned)</sub> model [66], potentially enhancing transfer learning effectiveness.

<sup>1</sup><https://www.kaggle.com/docs/notebooks>

<sup>2</sup><https://www.kaggle.com/docs/efficient-gpu-usage>

<sup>3</sup><https://github.com/chris-official/PyTorchGAF>

#### D. ESA-ADB

For our experiments, we utilized the European Space Agency Benchmark for Anomaly Detection in Satellite Telemetry (ESA-ADB) [1], the first large-scale, real-life satellite telemetry dataset with curated anomaly annotations. This dataset represents a significant advancement in spacecraft telemetry analysis, providing researchers with authentic operational data to develop and benchmark anomaly detection models.

The ESA Anomalies Dataset (ESA-AD) comprises telemetry data from three distinct ESA missions, with two of them included in the ESA-ADB. These missions represent diverse spacecraft types, orbits, and operational profiles. The telemetry spans a significant period, with an anonymized duration of approximately 17.5 years in total.

A key strength of this dataset is its comprehensive annotation scheme. The dataset includes 844 annotated events across two missions used in the benchmark, with 148 classified as anomalies. These events were meticulously labeled by spacecraft operations engineers and machine learning experts, and cross-verified using state-of-the-art algorithms.

The dataset is structured as a multivariate time series with the following characteristics:

- Complex characteristics including varying sampling frequencies across time and channels
- Data gaps caused by idle states and communication problems
- Trends connected with degradation of spacecraft components
- Concept drifts related to different operational modes and mission phases
- Diverse channel types including physical measurements, categorical status flags, counters, and binary telecommands

The dataset contains 176 channels in total (76 for Mission 1 and 100 for Mission 2), with 105 designated as target channels for anomaly detection and 71 as non-target channels to support the detection process. The total volume exceeds 1.55 billion data points, comprising over 7 gigabytes of compressed data.

Unlike previous spacecraft telemetry datasets, the ESA-AD maintains real operational characteristics while addressing privacy concerns through anonymization processes that preserve data integrity. This preservation of authentic data patterns is particularly valuable for anomaly detection research, as it allows evaluation of model performance against the actual complexities and challenges of space operations.

To support reproducible research, the dataset includes standardized train/validation/test splits and a comprehensive evaluation pipeline with new metrics designed specifically for satellite telemetry according to the latest advancements in time series anomaly detection.

For our experiments, we focused on the lightweight subset of Mission 1 since according to [1] it is

- suitable for potential on-board applications, challenging for algorithms,

- interesting for spacecraft operations engineers,
- relatively easy to visualize and analyze manually and
- not strongly dependent on other channels or subsystems.

#### E. Evaluation Metrics

To evaluate the performance of our anomaly detection models, we adopted the Corrected Event-wise  $F_{0.5}$ -score ( $CEF_{0.5}$ ) [1] as recommended in the ESA-ADB [1]. This specialized metric addresses several critical challenges inherent to spacecraft telemetry anomaly detection that are not adequately captured by traditional point-wise metrics.

Unlike conventional F1 scores that treat each data point independently, the  $CEF_{0.5}$  operates at the event level, addressing the temporal structure of spacecraft anomalies. This approach recognizes that anomalies in spacecraft telemetry typically manifest as continuous segments rather than isolated points, and a model's practical utility depends on detecting these anomalous events rather than maximizing the number of correctly classified individual data points. This builds on the anomaly-based scoring approach followed by [2], [47].

The metric incorporates three key innovations tailored to spacecraft telemetry evaluation:

- 1) **Event-based evaluation:** Rather than treating each time point independently, the metric first converts both predicted and ground truth anomalies into discrete events. An event is defined as a continuous sequence of anomalous points, where events separated by less than a predefined tolerance window (60 seconds in our implementation) are merged into a single event.
- 2) **Directional tolerance:** The metric implements an adjustable timing tolerance that accounts for early detection, recognizing that in operational scenarios, detecting an anomaly slightly before its actual occurrence can be as valuable as detecting it precisely when it occurs.
- 3) **Precision emphasis:** The  $CEF_{0.5}$  variant weights precision higher than recall (by a factor of 2), reflecting the operational reality that false alarms are particularly costly in spacecraft operations, where they may trigger unnecessary interventions that consume limited resources.

$CEF_{0.5}$  is based on precision and recall which are determined at the event level. An event is considered correctly detected if there is at least one predicted anomalous point that temporally overlaps with the ground truth event after applying the tolerance window.

This metric provides a more operationally relevant assessment of anomaly detection performance in spacecraft telemetry than traditional point-wise metrics, aligning our evaluation with the practical needs of mission operations where timely event detection with minimal false alarms is paramount.

#### F. Optimization for Edge Devices

We employed multi-objective optimization to balance the critical tradeoff between model accuracy and computational efficiency for spacecraft edge deployment. Rather than using  $CEF_{0.5}$  as our accuracy metric, we selected validation loss as

our primary performance indicator since it provides a more stable optimization target that does not depend on threshold selection, while also being computationally efficient to calculate during training.

For model complexity assessment, we utilized Multiply-Accumulate operations (MACs) as a proxy metric, which has become a standard practice in hardware-aware neural architecture search [38], [67]–[70]. The  $\mu$ NAS framework authors observed a strong linear correlation between MACs and actual microcontroller unit inference latency, validating its use as an accurate approximation for deployment performance [71]. While some researchers have criticized MACs as potentially inaccurate [67], [72]–[74], more precise alternatives typically require physical access to target hardware and introduce substantial evaluation overhead, making them impractical for our optimization process. Direct measurements on target devices would significantly slow the optimization cycle [67], [72].

Our optimization specifically targeted space-qualified computing systems which typically offer just 10-100x less processing power, memory, and storage compared to their terrestrial counterparts [3]. These substantial constraints are further compounded by radiation hardening requirements that often necessitate the use of older, more robust but less powerful semiconductor technologies [8]. Unlike terrestrial edge devices, spacecraft computing hardware cannot be upgraded once deployed, making efficient resource utilization particularly critical throughout the mission lifetime.

To accelerate our optimization process while preserving data diversity, we trained on 2 million samples from the 7 million available training samples, using sample weights to ensure exposure to all anomalous sequences. We standardized training at 2,048 steps per model configuration to ensure fair comparisons, with evaluations performed every 256 steps on the validation dataset (approximately 265,000 samples)—substantially more efficient than using the 7 million sample test dataset while avoiding potential test set overfitting.

Our optimization framework utilized Optuna’s Tree-structured Parzen Estimator (TPE) sampler to efficiently balance exploration and exploitation of the search space [40]. Each optimization run adjusted key architectural parameters including layer counts, filter sizes, and activation functions to achieve optimal performance within strict computational constraints. We fixed learning rate and batch size parameters to ensure performance differences resulted from architectural improvements rather than training hyperparameters. After training more than 50 model configurations for each approach, we identified *Pareto-optimal* configurations following the principles established by [36], where a solution is considered *Pareto-optimal* if it cannot be improved in one objective (accuracy) without degrading another (computational efficiency). This approach yielded a frontier of non-dominated solutions that represented the best possible tradeoffs between our competing objectives, allowing mission designers to select implementations according to specific hardware constraints. Importantly, we pruned configurations exceeding our *baseline* model size to focus computational resources on promising

lightweight alternatives suitable for spacecraft deployment.

For comprehensive model analysis, we utilized the MLTK Model Profiler to generate detailed insights into layer-wise computational demands and memory requirements [42]. This specialized profiler is particularly valuable for space applications as it provides accurate estimates of peak memory usage, MACs distribution across layers, and overall model efficiency on resource-constrained hardware. The insights gained from this profiling guided our strategic decisions in architecture modification, helping identify specific layers and components that could be optimized without compromising detection performance.

## IV. EXPERIMENTS AND RESULTS

### A. Results of Forecasting & Threshold, Direct Classification and Image Classification

Table I presents the comparative performance of our *baseline* and *optimized* anomaly detection approaches. Among the *baseline* models, the *forecasting & threshold* approach demonstrated superior detection capability with the highest CEF<sub>0.5</sub> of 92.7%, substantially outperforming both *direct classification* (72.4%) and *image classification* (48.6%). This performance advantage stems primarily from its exceptional precision (99.8%) combined with reasonable recall (72.3%), indicating that when this approach identifies an anomaly, it is almost certainly correct—a particularly valuable characteristic for spacecraft operations where false alarms can trigger costly interventions.

*Direct classification* achieved high precision (95.3%) but struggled with recall (36.9%), suggesting that while it rarely generates false alarms, it misses a significant portion of actual anomalies. This limitation likely stems from the approach’s inability to leverage temporal context beyond its fixed window size, making subtle deviations difficult to detect. Despite this limitation, it is worth noting that our *direct classification* implementation still outperforms most benchmark models [1] on the ESA-ADB in terms of CEF<sub>0.5</sub>, demonstrating its viability as an anomaly detection approach. This precision-oriented performance aligns with spacecraft operators’ preferences, who typically consider false alarms more problematic than missed anomalies, as they can rely on spacecraft autonomy for basic protection while avoiding unnecessary interventions. This operational reality reinforces existing research indicating that minimizing false positives is a critical consideration in spacecraft operations [1], [2].

The *image classification* approach, despite its theoretical advantages in spatial pattern recognition, performed surprisingly poorly with *baseline* models, showing the lowest CEF<sub>0.5</sub> (48.6%) and precision (50.2%). This underperformance may be attributed to information loss during the GAF transformation process, where temporal relationships are encoded into spatial patterns that may not be effectively interpreted by the model without extensive training data. Interestingly, however, the recall of 43.1% slightly exceeds that of the *direct classification* approach, suggesting potential complementarity between these methods.

TABLE I: Performances of Baseline and Optimized Anomaly Detection Systems

Metric	Baseline			Optimized		
	<i>Direct classification</i>	<i>Image classification</i>	<i>Forecasting &amp; threshold</i>	<i>Direct classification</i>	<i>Image classification</i>	<i>Forecasting &amp; threshold</i>
Event-wise Precision (%)	95.3	50.2	99.8	91.1	99.1	99.9
Event-wise Recall (%)	36.9	43.1	72.3	16.9	32.3	61.6
Event-wise $F_{0.5}$ (%)	72.4	48.6	<b>92.7</b>	48.5	70.1	88.8
Parameters	310k	1,254k	317k	<b>19k</b>	128k	29k
MACs	56,538k	959,367k	56,735k	907k	45,000k	<b>354k</b>
RAM (KB)	1,606	8,193	2,043	122	1,024	<b>59</b>
ROM (KB)	1,268	4,902	1,294	<b>149</b>	508	166

From a computational perspective, both *direct classification* and *forecasting & threshold* demonstrated comparable resource requirements, with approximately 310k-317k parameters and 56M-57M MACs. Despite their similar computational footprints, the *forecasting & threshold* approach achieved significantly better detection performance, suggesting superior algorithmic efficiency for spacecraft telemetry analysis. The *image classification* approach, with over 1.25 million parameters and 959M MACs, proved substantially more resource-intensive while delivering the poorest detection results, making it the least efficient option for space-constrained deployment scenarios.

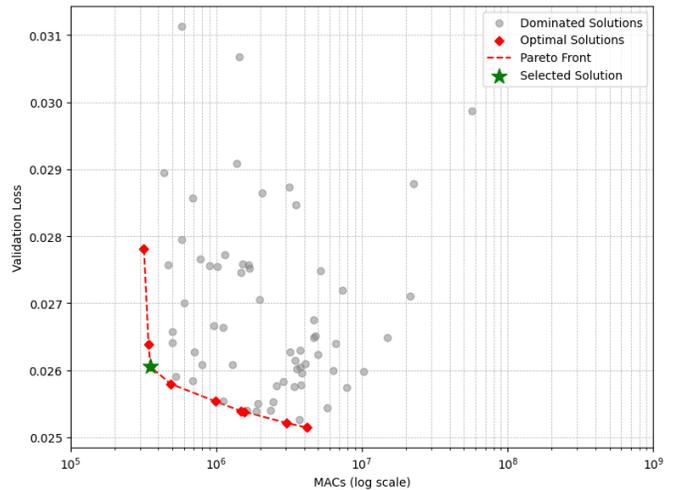
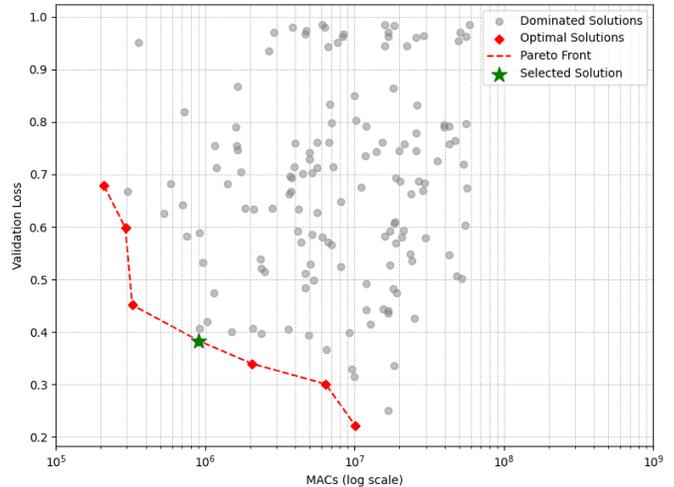
Memory utilization patterns further highlight the implementation challenges of these approaches. *Image classification* required approximately 8,193 KB of RAM—four times more than the other approaches—due to the memory-intensive nature of image transformation and processing. This memory footprint would make deployment impractical on many spacecraft computing platforms that typically offer limited RAM resources. *Direct classification* and *forecasting & threshold* demonstrated similar memory profiles (1,606 KB and 2,043 KB of RAM, respectively), aligning with their comparable parameter counts.

These *baseline* results indicate that *forecasting & threshold* offers the best combination of detection capability and computational efficiency for spacecraft telemetry anomaly detection. Its superior  $CEF_{0.5}$ , coupled with reasonable resource requirements, makes it particularly well-suited for spacecraft deployment scenarios where both accurate anomaly detection and computational efficiency are critical. *Direct classification* presents a viable alternative with good precision but limited recall, while *baseline image classification* appears unsuitable for spacecraft deployment without significant optimization, due to its poor detection performance and excessive resource requirements.

### B. Optimization for Edge Devices

1) *Pareto-Optimal Architecture Selection*: The core challenge of deploying deep learning models on spacecraft hardware lies in balancing detection accuracy against severe computational constraints. To systematically explore this tradeoff, we applied multi-objective optimization using the *Pareto-optimal* approach, which identifies solutions where one objective cannot be improved without degrading another. Figures 1,

2, and 3 visualize this optimization process for our three anomaly detection approaches.


 Fig. 1: Pareto Front Plot for *Forecasting & Threshold*.

 Fig. 2: Pareto Front Plot for *Direct Classification*.

In each figure, the horizontal axis represents computational complexity measured in MACs, while the vertical axis shows validation loss, our proxy for model accuracy. The

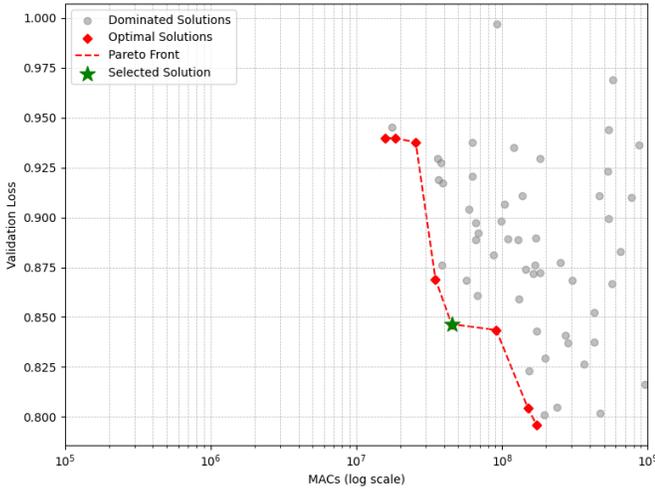


Fig. 3: Pareto Front Plot for *Image Classification*.

gray dots represent candidate model configurations evaluated during our optimization process. These *dominated* solutions—configurations for which there exists at least one alternative that is superior in at least one objective without being inferior in the other—represent inefficient designs that should be avoided in resource-constrained environments.

The red dots mark the *Pareto-optimal* configurations that form the *Pareto-optimal*—models that achieve the best possible validation loss for a given computational budget or, equivalently, the lowest computational cost for a given level of accuracy. These non-dominated solutions represent the most efficient implementations of each detection approach, providing spacecraft engineers with a range of deployment options calibrated to specific mission constraints.

For our final model selection, we prioritized balanced performance by choosing configurations closest to the origin (marked as a green dot), sometimes referred to as the “knee point” of the *Pareto* curve. This selection criterion identifies models that deliver substantial reductions in computational requirements while maintaining acceptable detection performance—a critical consideration for spacecraft deployment where both factors significantly impact mission success.

The *Pareto* plots reveal distinct optimization characteristics across our three detection approaches. The *forecasting & threshold* approach (Figure 1) demonstrates a relatively gradual tradeoff curve, suggesting that detection performance can be maintained even with significant reductions in model complexity. The *direct classification* approach (Figure 2) exhibits a steeper *Pareto-optimal*, indicating a more sensitive relationship between computational resources and detection capability. The *image classification* optimization (Figure 3) shows the most challenging tradeoff landscape, with substantial performance penalties accompanying even moderate reductions in computational resources.

These optimization results informed our final model selection, guiding us toward implementations that maintain critical detection capabilities while operating within the severe

computational constraints of spacecraft edge devices. The selected configurations, marked on each *Pareto-optimal*, were subsequently evaluated on the complete test dataset to verify their real-world performance, as detailed in the following sections.

### C. Resource Usage in On-Board Computing Environments

To assess the practical deployability of our *optimized* models, we analyzed their resource requirements in the context of two representative spacecraft computing platforms: the resource-constrained CubeSat with its PIC32MZ M14K processor (200 MHz), 16 MB of RAM and 64 MB of flash memory, and the more capable OPS-SAT experimental platform with an Altera Cyclone V SoC (ARM dual-core 800 MHz Cortex-A9), 1 GB DDR3 RAM and 8 GB industrial-grade SD card storage. Table II quantifies our models’ requirements as absolute values and as percentages of available resources on each platform.

The *baseline* implementations demonstrate why direct deployment of sophisticated anomaly detection models on spacecraft hardware has been challenging. While all *baseline* models could technically fit within OPS-SAT’s generous memory allocation, they would consume significant resources—particularly the *image classification* approach requiring 8,193 KB of RAM (0.78% of OPS-SAT capacity). More concerning is their deployment on CubeSat-class hardware, where the *baseline image classification* would consume over 50% of available RAM, leaving insufficient memory for other critical spacecraft functions. Even the more modest *direct classification* and *forecasting & threshold* approaches would claim approximately 10% and 12% of CubeSat RAM respectively—substantial allocations for a single subsystem.

Our optimization efforts dramatically transformed this landscape. The *optimized forecasting & threshold* implementation now requires just 59 KB of RAM—a 34.6 $\times$  reduction from its *baseline*—representing a mere 0.36% of CubeSat memory and an almost negligible 0.01% of OPS-SAT resources. This exceptional efficiency preserves the highest detection performance (CEF<sub>0.5</sub> score of 88.8%) among our *optimized* approaches. Similarly, the *optimized direct classification* model achieved a 13.2 $\times$  reduction in RAM usage to just 122 KB, requiring only 0.74% of CubeSat memory. Both models demonstrate storage requirements well within spacecraft constraints, with Read-Only Memory (ROM) requirements of 166 KB and 149 KB respectively, each using less than 0.3% of CubeSat storage capacity.

The *image classification* approach, while significantly improved with a 8.0 $\times$  reduction in RAM usage, still requires 1,024 KB (1 MB) of memory. While this represents a reasonable 6.25% of CubeSat RAM, it remains substantially higher than our other approaches. This memory requirement, combined with its more intensive computational demands, makes *image classification* less suitable for deployment on severely constrained platforms despite its impressive detection precision.

All three *optimized* approaches easily meet the resource constraints of the OPS-SAT platform, with each requiring less than 0.1% of available RAM and negligible portions of storage capacity. This indicates that more capable spacecraft computing platforms can readily support multiple concurrent anomaly detection systems, potentially enabling ensemble approaches that leverage the complementary strengths of different detection methods.

These results demonstrate that our neural architecture optimization approach successfully bridges the gap between sophisticated anomaly detection capabilities and the severe resource constraints of spacecraft computing environments. The *optimized forecasting & threshold* and *direct classification* models are particularly notable, as both satisfy common microcontroller deployment criteria (<250 KB RAM, <250 KB ROM) while requiring minimal computational resources. The *forecasting & threshold* approach represents the most balanced solution, with near-*baseline* detection performance packed into an extraordinarily efficient implementation suitable for even the most constrained spacecraft computing platforms.

Our optimization efforts yielded remarkable reductions in computational requirements across all three approaches, as shown in the “*optimized*” columns of Table I. These *optimized* models correspond to the solutions highlighted with green dots in Figures 1, 2, and 3, representing our selected configurations from the respective *Pareto fronts*.

The *forecasting & threshold* approach maintained the highest overall detection performance with a  $CEF_{0.5}$  of 88.8% while achieving dramatic resource reductions. Most notably, this *optimized* implementation requires just 354k MACs—a 99.4% reduction from its *baseline* counterpart—while preserving near-perfect precision (99.9%). Despite a moderate decrease in recall from 72.3% to 61.6%, the model maintained high operational utility while reducing its RAM footprint by 97.1% to just 59 KB, making it deployable even on highly constrained spacecraft computing platforms.

The *direct classification* approach experienced the most significant performance trade-off, with its  $CEF_{0.5}$  decreasing from 72.4% to 48.5%. This substantial drop resulted primarily from reduced recall (from 36.9% to 16.9%), though precision remained relatively high at 91.1%. This performance compromise may be attributed to two factors: first, our optimization targeted the  $CEF_{0.5}$  metric which heavily weights precision over recall, potentially leading to threshold settings that favor precision; second, this approach underwent the most aggressive parameter reduction (16.4 $\times$ ), potentially limiting model capacity. However, this compromise enabled extraordinary computational efficiency gains, with parameters reduced to just 19K and MACs decreased by 98.4% to 907k. The model’s memory requirements shrank dramatically to just 122 KB of RAM and 149 KB of ROM—the smallest storage footprint among all *optimized* implementations—making it potentially suitable for extremely resource-constrained deployment scenarios where detection of some anomalies is preferable to none.

Perhaps most surprising was the optimization of the *image*

*classification* approach, which not only reduced computational demands but actually improved detection performance. The  $CEF_{0.5}$  increased from 48.6% to 70.1%, driven by a dramatic improvement in precision from 50.2% to 99.1%. While recall decreased moderately from 43.1% to 32.3%, the near-elimination of false positives represents a significant operational advantage. These improvements accompanied substantial resource reductions, with parameters decreased by 89.8% to 128K and MACs reduced by 95.3% to 45M. Despite these gains, the *image classification* approach still requires significantly more computational resources than the other methods, although the near-elimination of false positives makes it worth considering for critical applications where false alarms are extremely costly.

These optimization results demonstrate that neural architecture optimization can dramatically reduce the computational requirements of spacecraft anomaly detection systems while maintaining operational utility. The *optimized forecasting & threshold* implementation represents the most balanced solution, with high detection performance and minimal resource requirements. At just 59 KB of RAM and 354k MACs, this model is deployable on most spacecraft computing platforms, including severely constrained CubeSat systems. While the other approaches offer viable alternatives for specific operational contexts—*direct classification* for absolute minimal resource usage and *image classification* for extremely high precision—the *optimized forecasting & threshold* approach provides the best overall combination of detection capability and computational efficiency for spacecraft telemetry anomaly detection.

## V. CONCLUSION AND FUTURE WORK

### A. Conclusion

This research demonstrates that sophisticated deep learning-based anomaly detection capabilities can be successfully deployed on space-grade edge devices through systematic neural architecture optimization. Our experiments with three distinct anomaly detection approaches—*forecasting & threshold*, *direct classification*, and *image classification*—reveal important trade-offs between detection performance and computational efficiency in the context of spacecraft telemetry analysis.

The *forecasting & threshold* approach consistently delivered the best balance of detection capability and resource efficiency, achieving a 92.7%  $CEF_{0.5}$  in its *baseline* form and maintaining 88.8% after optimization while reducing computational requirements by over 99%. This approach represents a compelling solution for on-board anomaly detection, requiring just 59 KB of RAM (0.36% of CubeSat memory) while maintaining near-perfect precision. However, its accuracy depends heavily on a sophisticated non-parametric dynamic thresholding mechanism that operates on batches of data, introducing computational overhead beyond the model itself.

The *direct classification* approach offers unique advantages despite its lower  $CEF_{0.5}$ . As the most compact implementation at just 19K parameters and 149 KB ROM, it produces interpretable anomaly probabilities that can be converted

TABLE II: Resource Usage of Baseline and Optimized Anomaly Detection Systems Relative to On-Board Resources

Metric	Baseline			Optimized		
	<i>Direct classification</i>	<i>Image classification</i>	<i>Forecasting &amp; threshold</i>	<i>Direct classification</i>	<i>Image classification</i>	<i>Forecasting &amp; threshold</i>
Event-wise $F_{0.5}$ (%)	72.4	48.6	<b>92.7</b>	48.5	70.1	88.8
RAM (KB)	1,606	8,193	2,043	122	1,024	<b>59</b>
ROM (KB)	1,268	4,902	1,294	<b>149</b>	508	166
RAM (%) CubeSat (16 MB)	9.80	50.01	12.47	0.74	6.25	<b>0.36</b>
ROM (%) CubeSat (64 MB)	1.93	7.48	1.97	<b>0.23</b>	0.78	0.25
RAM (%) OPS-SAT (1 GB)	0.15	0.78	0.19	0.01	0.10	<b>0.01</b>
ROM (%) OPS-SAT (8 GB)	0.02	0.06	0.02	< <b>0.01</b>	0.01	< 0.01

to binary classifications with minimal post-processing. This characteristic, combined with its high precision (91.1%) and ability to operate without batch-dependent thresholds, makes it particularly suitable for real-time detection in environments with limited computational capabilities.

While *image classification* achieved competitive accuracy after optimization—increasing its  $CEF_{0.5}$  from 48.6% to 70.1%—it comes with higher computational costs that limit its deployment options on the most constrained platforms. Nevertheless, its distinct approach offers complementary strengths that could be valuable in ensemble systems.

Our multi-objective optimization approach using the *Pareto-optimal* methodology provides spacecraft engineers with a systematic framework for balancing detection performance against hardware constraints, enabling deployment of sophisticated anomaly detection even on platforms with severe computational limitations.

### B. Future Work

Future research should explore several promising directions. First, investigating hybrid or ensemble methods that combine the complementary strengths of multiple detection approaches could potentially improve overall detection robustness without significantly increasing resource requirements. The high precision of *forecasting & threshold* coupled with the real-time processing advantages of *direct classification* suggests particular promise in this area.

Second, addressing the remaining detection limitations in *direct* and *image classification* through more advanced thresholding mechanisms could further enhance performance without compromising computational efficiency. Additionally, extending our multi-objective optimization framework to include additional spacecraft-relevant objectives such as energy consumption and inference latency would provide a more comprehensive optimization landscape.

Third, comparing our neural architecture optimization approach with other model compression techniques like quantization, pruning, and knowledge distillation could reveal complementary pathways to efficiency. Scaling our solution to the full dataset of Mission 1 and Mission 2 would further generalize our findings across diverse spacecraft operational profiles.

Finally, integrating these anomaly detection systems with autonomous recovery and fault-management subsystems would enable comprehensive evaluation under realistic mission conditions, particularly through opportunities like ESA's OPS-SAT platform, providing invaluable validation of these approaches in authentic operational environments.

### REFERENCES

- [1] K. Kotowski, C. Haskamp, J. Andrzejewski, B. Ruszczak, J. Nalepa, D. Lakey, P. Collins, A. Kolmas, M. Bartesaghi, J. Martinez-Heras, and G. D. Canio, "European Space Agency Benchmark for Anomaly Detection in Satellite Telemetry," 2024. [Online]. Available: <https://arxiv.org/abs/2406.17826>
- [2] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding," 2018.
- [3] G. Lentaris, K. Maragos, I. Stratakos, L. Papadopoulos, O. Papanikolaou, D. Soudris, M. Lourakis, X. Zabulis, D. Gonzalez-Arjona, and G. Furano, "High-Performance Embedded Computing in Space: Evaluation of Platforms for Vision-Based Navigation," *Journal of Aerospace Information Systems*, vol. 15, no. 4, pp. 178–192, 2018.
- [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] 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.
- [6] R. Dyer, R. Porta, A. Loretucci, E. Valido, F. Ares, and F. Perlik, "Metop-sg operations concept: From spacecraft operations to configuration of onboard autonomy," in *2018 SpaceOps Conference*. American Institute of Aeronautics and Astronautics, May 2018. [Online]. Available: <http://dx.doi.org/10.2514/6.2018-2604>
- [7] M. Woods, L. Baldwin, G. Wilson, S. Hall, R. Aylett, D. Long, R. Vitulli, and A. Pidgeon, "Mmops: Assessing the impact of on-board autonomy for deep space robotic missions," in *SpaceOps 2006 Conference*. American Institute of Aeronautics and Astronautics, Jun. 2006. [Online]. Available: <http://dx.doi.org/10.2514/6.2006-5977>
- [8] G. Furano, G. Meoni, A. Dunne, D. Moloney, V. Ferlet-Cavrois, A. Tavoularis, J. Byrne, L. Buckley, M. Povey, D. Caroselli *et al.*, "Towards the Use of Artificial Intelligence on the Edge in Space Systems: Challenges and Opportunities," in *2020 IEEE Aerospace Conference*. IEEE, 2020, pp. 1–12.
- [9] J. Heras and A. Donati, "Enhanced Telemetry Monitoring with Novelty Detection," *AI Magazine*, vol. 35, no. 4, pp. 37–46, 2014.
- [10] 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>

- [11] 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.
- [12] Z. Zamanzadeh Darban, "Deep Learning for Time Series Anomaly Detection: A Survey," *ACM Computing Surveys*, vol. 56, no. 4, 2023. [Online]. Available: <https://dl.acm.org/doi/10.1145/3691338>
- [13] 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>
- [14] 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>
- [15] L. Yang, Y. Ma, F. Zeng, X. Peng, and D. Liu, "Improved Deep Learning Based Telemetry Data Anomaly Detection to Enhance Spacecraft Operation Reliability," *Microelectronics Reliability*, vol. 126, p. 114311, Nov. 2021. [Online]. Available: <https://doi.org/10.1016/j.microrel.2021.114311>
- [16] 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.
- [17] 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>
- [18] F. Harrell, "Classification vs. Prediction," Jan 2017. [Online]. Available: <https://www.fharrell.com/post/classification>
- [19] Z. Wang, W. Yan, and T. Oates, "Time Series Classification from Scratch with Deep Neural Networks: A Strong Baseline," 2016.
- [20] 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>
- [21] 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>
- [22] 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>
- [23] 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>
- [24] D. Lakey and T. Schlippe, "Anomaly Detection in Spacecraft Telemetry: Forecasting vs. Classification," in *IEEE Space Computing Conference (SCC)*. Mountain View, CA, USA: IEEE, July 2024.
- [25] 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>
- [26] 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>
- [27] 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>
- [28] 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>
- [29] —, "Spatially Encoding Temporal Correlations to Classify Temporal Data Using Convolutional Neural Networks," 2015.
- [30] 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-b043becbdfb3>
- [31] 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.
- [32] 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>
- [33] SatCatalog, "IMT - CubeSat On-Board Computer - Datasheet," 2021, accessed on July 8, 2021. [Online]. Available: [https://satcatalog.s3.amazonaws.com/components/458/SatCatalog\\_-\\_IMT\\_-\\_CubeSat\\_On-Board\\_Computer\\_-\\_Datasheet.pdf](https://satcatalog.s3.amazonaws.com/components/458/SatCatalog_-_IMT_-_CubeSat_On-Board_Computer_-_Datasheet.pdf)
- [34] D. J. Evans, "OPS-SAT: Operational concept for ESA's first mission dedicated to operational technology," in *SpaceOps 2016 Conference*. American Institute of Aeronautics and Astronautics, May 2016.
- [35] D. Evans, M. Kubicka, C. Cherciu, and C. Coelho, "Ops-sat experimenter," in *European Ground System Architecture Workshop (ESAW) 2017*. European Space Agency, 2017. [Online]. Available: [https://web.archive.org/web/20171223042533/http://www.esa-esaw2017.eu/sites/default/files/OPS-SAT\\_Experimenter\\_Dave\\_Evans\\_Final%20.pdf](https://web.archive.org/web/20171223042533/http://www.esa-esaw2017.eu/sites/default/files/OPS-SAT_Experimenter_Dave_Evans_Final%20.pdf)
- [36] Y. Jin and B. Sendhoff, "Pareto-Based Multiobjective Machine Learning: An Overview and Case Studies," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 3, pp. 397–415, 2008.
- [37] C. Liu, Y. Cai, H. Zhang, X. Xu, Q. Wan, and C. Song, "MicroNAS: Neural Architecture Search for Efficient Deep Learning on Microcontrollers," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 43, no. 1, pp. 185–198, 2024.
- [38] C. Banbury, C. Zhou, I. Fedorov, R. Matas, U. Thakker, D. Gope, V. Janapa Reddi, M. Mattina, and P. Whatmough, "MicroNets: Neural Network Architectures for Deploying TinyML Applications on Commodity Microcontrollers," in *Proceedings of Machine Learning and Systems*, vol. 3, 2021, pp. 517–532.
- [39] E. George, X. Shen, J. Rodriguez, X. Huie, K. Kouzbari, and R. Haider, "Spacecraft Anomaly Detection Using Variational Autoencoders," *Journal of Aerospace Computing, Information, and Communication*, vol. 17, no. 1, pp. 27–39, 2020.
- [40] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-Generation Hyperparameter Optimization Framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2623–2631.
- [41] S. Brockmann and T. Schlippe, "Optimizing Convolutional Neural Networks for Image Classification on Resource-Constrained Microcontroller Units," *Computers*, vol. 13, no. 7, p. 173, 2024.
- [42] Silicon Labs, "MLTK Model Profiler," [https://siliconlabs.github.io/mltk/docs/guides/model\\_profiler.html](https://siliconlabs.github.io/mltk/docs/guides/model_profiler.html), 2023, accessed: 2024-07-01.
- [43] D. Lakey, "A Comparison of Deep Learning Architectures for Spacecraft Anomaly Detection," 2024. [Online]. Available: <https://zenodo.org/doi/10.5281/zenodo.10829339>
- [44] 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>
- [45] P. Fortescue, G. Swinerd, and J. Stark, *Spacecraft Systems Engineering*, 4th ed. Nashville, TN: John Wiley & Sons, 2011.
- [46] 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.
- [47] 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>
- [48] 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>
- [49] 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>
- [50] 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>

- [51] 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>
- [52] 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>
- [53] 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>
- [54] 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>
- [55] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2015.
- [56] 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>
- [57] 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>
- [58] 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>
- [59] 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>
- [60] Pytorch, "Models and Pre-trained Weights; Torchvision 0.17 Documentation - pytorch.org," <https://pytorch.org/vision/stable/models.html>, [Accessed 06-02-2024].
- [61] 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>
- [62] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 2014. [Online]. Available: <https://arxiv.org/abs/1409.1556>
- [63] 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>
- [64] 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>
- [65] 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](http://dx.doi.org/10.1007/978-3-030-86340-1_11)
- [66] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2015. [Online]. Available: <https://arxiv.org/abs/1512.03385>
- [67] H. Benmeziane, K. El Maghraoui, H. Ouarnoughi, S. Niar, M. Wistuba, and N. Wang, "A Comprehensive Survey on Hardware-Aware Neural Architecture Search," *arXiv preprint arXiv:2101.09336*, 2021. [Online]. Available: <http://arxiv.org/pdf/2101.09336v1>
- [68] A. M. Garavagno, E. Ragusa, A. Frisoli, and P. Gastaldo, "An Affordable Hardware-Aware Neural Architecture Search for Deploying Convolutional Neural Networks on Ultra-Low-Power Computing Platforms," *IEEE Sensors Letters*, vol. 8, no. 5, pp. 1–4, 2024.
- [69] C.-H. Hsu, S.-H. Chang, J.-H. Liang, H.-P. Chou, C.-H. Liu, S.-C. Chang, J.-Y. Pan, Y.-T. Chen, W. Wei, and D.-C. Juan, "MONAS: Multi-Objective Neural Architecture Search using Reinforcement Learning," *CoRR*, 2018. [Online]. Available: <http://arxiv.org/pdf/1806.10332v2>
- [70] Y. Yang, J. Zhan, W. Jiang, Y. Jiang, and A. Yu, "Neural Architecture Search for Resource Constrained Hardware Devices: A Survey," *IET Cyber-Physical Systems: Theory & Applications*, vol. 8, no. 3, pp. 149–159, 2023.
- [71] E. Liberis, Dudziak, and N. D. Lane, "NAS: Constrained Neural Architecture Search for Microcontrollers," *arXiv preprint arXiv:2010.14246*, 2020. [Online]. Available: <http://arxiv.org/pdf/2010.14246v3>
- [72] T. King, Y. Zhou, T. Röddiger, and M. Beigl, "Micronas for Memory and Latency Constrained Hardware Aware Neural Architecture Search in Time Series Classification on Microcontrollers," *Scientific Reports*, vol. 15, no. 1, p. 7575, 2025.
- [73] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "MnasNet: Platform-Aware Neural Architecture Search for Mobile," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2815–2823.
- [74] B. Wu, X. Dai, P. Zhang, Y. Wang, F. Sun, Y. Wu, Y. Tian, P. Vajda, Y. Jia, and K. Keutzer, "FBNet: Hardware-Aware Efficient ConvNet Design via Differentiable Neural Architecture Search," *arXiv preprint arXiv:1812.03443*, 2018. [Online]. Available: <http://arxiv.org/pdf/1812.03443v3>