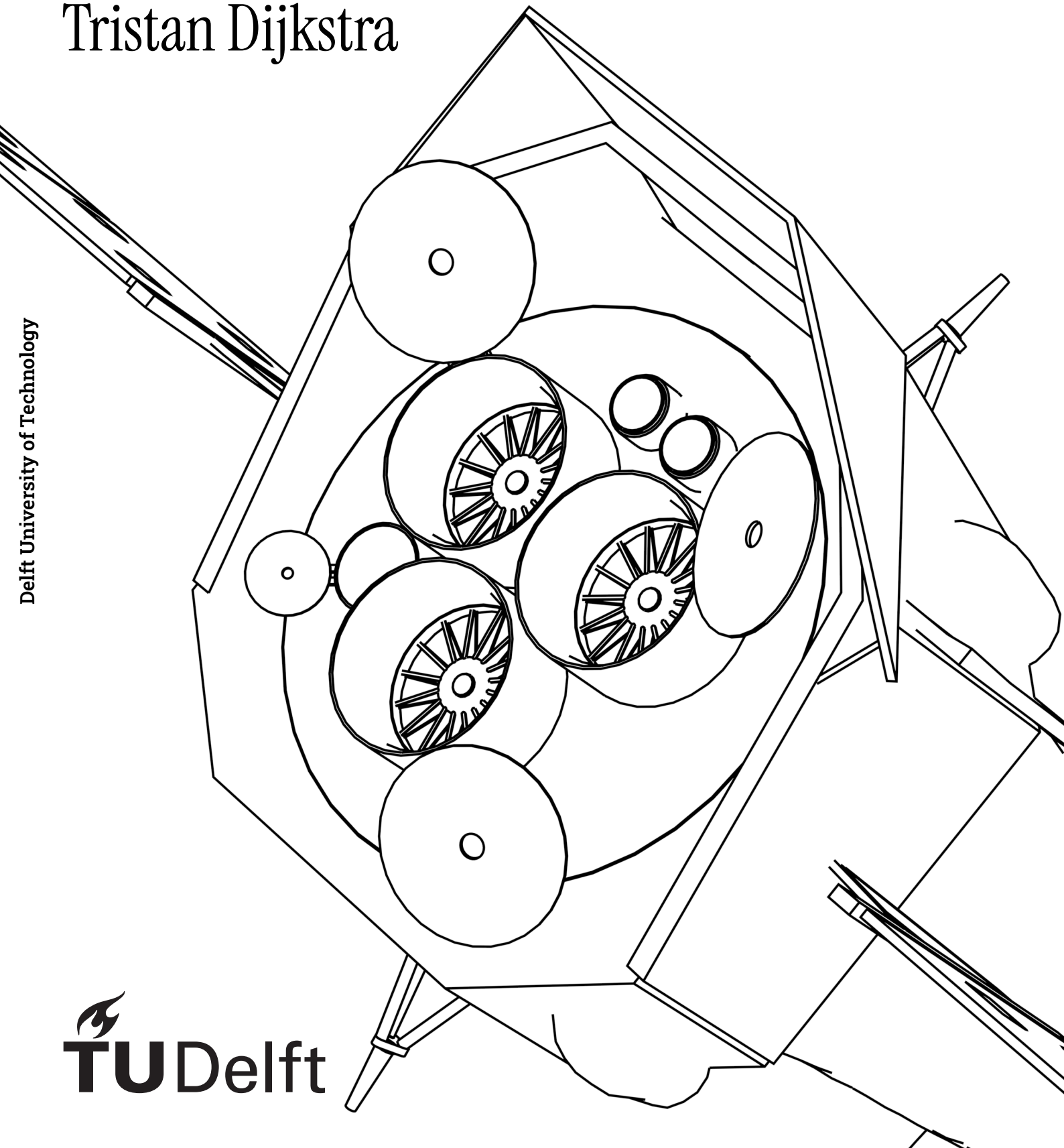
MSc Thesis

Machine Learning-based Anomaly Detection in XMM-Newton Telemetry Data

Tristan Dijkstra



Delft University of Technology

TU Delft

MSc Thesis

Machine Learning-based Anomaly Detection in XMM-Newton Instrument Telemetry Data

by

Tristan Dijkstra

to obtain the degree of

Master of Science in Aerospace Engineering

at the Delft University of Technology,

to be defended publicly on Monday May 12, 2025 at 14:00.

Student number:	4798139	
Project duration:	June, 2024 – March, 2025	
Thesis committee:	Dr. J. Guo,	TU Delft, Committee Chair
	Dr. Ir. R. Sabzevari,	TU Delft, External Examiner
	Dr. S. Speretta,	TU Delft, Supervisor
	Dr.-Ing. P. Gómez,	ESA ESAC, Supervisor

Cover: Stylised illustration of XMM-Newton. Own work using 3D model obtained from https://scifleet.esa.int/

Code: Once cleared, the code developed for this thesis will become available here: https://github.com/tristandijkstra/xmm_anomaly_thesis

TU Delft

Preface

The completion of this thesis marks an end to my six-and-a-half-year journey at TU Delft as well as multiple decades of studying. Looking back, I can still vividly remember the day I was accepted into the Aerospace Bachelor's, and I am reminded of the challenges, growth, and incredible memories that I have experienced since then. I feel incredibly fortunate to be able to finish my studies with a thesis on a topic which I have been interested in for some years, and I am thankful to a number of people who have supported me along the way.

I would like to start by extending my deepest gratitude to my two supervisors, Stefano Speretta and Pablo Gómez, for the opportunity, for your guidance and for your patience. At times, it felt like the thesis was going nowhere, yet you both kept kept optimistic outlook, and rightfully so. Beyond the technical knowledge you have instilled in me, you have both significantly influenced my way of thinking, which I hope to carry with me in my future endeavours.

From the XMM-Newton team at ESAC, I would like to thank Meeri Harkki, Landry Amiard, Pedro Calderón Riaño and Peter Kretschmar for their invaluable support during the project. Special thanks go to Landry and Meeri for regularly and enthusiastically assisting me with my many questions on XMM. From earlier in my studies at TU Delft, I would also like to thank Junzi Sun for sparking my interest in data science in a way only few teachers can.

I would like to thank my friends, both the old bunch, who have stuck with me since my final years in high school as well as those I have met in Delft. You have all had a lasting impression on me, and have helped shape me into the person I am today. A special thanks goes to my closest friend Rahim, who has been with me through it all.

Last but not least, I would like to thank my parents, my younger brother and my grandmother for always believing in me, for encouraging my education, and for their life-long and unwavering support, even from 8000 kilometres away. This thesis is dedicated to you.

Tristan Dijkstra
Delft, March 2025

Summary

To ensure their long-term safety, spacecraft transmit telemetry data from thousands of onboard sensors to Earth, which is then continuously monitored for anomalies. Ground operators are commonly supported by alarming systems, which actively monitor if telemetry goes beyond pre-defined thresholds, as well as statistical pattern matching systems to find recurring anomalies. However, anomaly detection remains an expensive, time-consuming, and labour-intensive task. With the number of satellites expected to increase in the coming years, a number of space agencies have been researching Machine Learning-based (ML) Time Series Anomaly Detection (TSAD) methods to increase automation. Such methods have already been applied successfully for cybersecurity and fraud detection. However, a shortage of quality spacecraft telemetry-related benchmark datasets inhibits adoption in the spacecraft operations.

XMM-Newton is a space telescope operated by the European Space Agency (ESA), that studies high-energy cosmic X-ray sources. The spacecraft was launched in 1999 and its ground operations have been significantly automated since then. Operators are now implementing machine learning-based methods to further facilitate operations as part of a broader push towards such methods at ESA. The spacecraft has been operational for over 25 years, providing a large span of data for an eventual anomaly benchmark.

In collaboration with the XMM-Newton team and the Data Science Section at the European Space Astronomy Centre (ESAC), this thesis explored machine learning-based anomaly detection methods applied to instrument telemetry data from the XMM-Newton space telescope. The primary research question was:

What is a suitable approach to construct a dataset of anomalies in XMM-Newton instrument telemetry data using ML-based TSAD techniques?

The project was focussed on two scientific instruments: the *pn* sensor of the European Photon Imaging Camera (EPIC-PN or PN for short) and the Optical Monitor (OM). Beginning with a data exploration phase, XMM telemetry was found to contain a number of challenging features, including: seasonal eclipses, and a high volume of missing data. The Darts Python library was chosen to simplify development, leading to a semi-supervised forecasting approach. The steps used to go from raw telemetry data to a catalogue of anomalies are described below:

1. **From raw telemetry data to machine learning-ready data** — To make it compatible with ML-based TSAD algorithms, raw telemetry data is processed to account for varying sample rates, differing magnitudes across channels, status and mode channels. Auxiliary data, like eclipses and orbital data are also added.
2. **From machine learning-ready data to forecasted telemetry data** — Semi-supervised forecasting anomaly detection methods rely on autoregression to reproduce a version of the telemetry data without anomalies. The forecasted telemetry data can then be compared to the original, revealing points with high forecasting error to be anomalies. In addition to the target telemetry channel being forecasted, the forecasting models are provided with a number of covariate channels containing real data, to provide additional context. These covariates are always presented as true data, meaning the approach is not true forecasting. Multiple forecasting models were tested, finding that more than one model has good performance but that the best model depends on the channel, chosen preprocessing steps and the amount of hyper-parameter tuning used. In the end, LSTM-based models were selected due to their relative simplicity, compatibility with a number of target channels and fast runtimes. To exclude anomalous segments from the training data, sample weights for these segments are set to zero.
3. **From forecasted telemetry data to anomaly detections** — The forecasting error can be modified in various ways to obtain an anomaly score. Various scorers were compared, ranging from the basic absolute forecasting error to unsupervised anomaly detection methods. A number of earlier detections as well as some ground-truth anomalies provided by the XMM team were used as a benchmark to compare scoring functions. None of the scorers tested could detect all of the benchmark anomalies, and in the end, a combination of multiple unsupervised algorithms are used: two Isolation Forest scorers and one Histogram-Based Outlier Scorer (HBOS), with window sizes, 75, 240 and 240 minutes respectively. A static quantile threshold is then be applied to retrieve anomaly detections.
4. **From anomaly detections to a catalogue of anomalies** — A number of post-processing steps are applied to refine the detections, introduce supplemental metrics and convert them to a catalogue of anomalies in tabular format.

A data pipeline is implemented to automatically go through each of the four steps, using a number of configuration files as input, and producing anomalies, metrics and intermediate data products as output. This pipeline was successfully run to three temperature-related target channels yielding a total of 40 detections after filtering out forecasting artefacts and recurring appearances of a recurring spike anomaly. Most of the detections extend prominently beyond the nominal behaviour of the channels, with only 2 inconspicuous detections. This can partly be attributed to the scoring system: each scorer is only fed with the forecasting error and applying a static quantile threshold yields only the anomalies with the largest forecasting error, thus exceeding furthest from the nominal behaviour. At the same time, the channels analysed provide little opportunity to detect inconspicuous anomalies: the first has a straight line as its nominal behaviour and the other two are highly discretised, containing only three possible values in their nominal range.

The detections were compared to a collection of anomaly reports provided by the XMM team, finding that only three detections have matching anomaly reports. At least one other relevant anomaly report yielded no detection due to deficiencies in the pipeline. Of the remaining detections, only three could be discussed with an instrument engineer, yet all three were found to be real anomalies, providing a positive indication of the remaining results. Testing the pipeline on a larger number of target channels and comparing resulting detections to a larger collection of existing anomalies would provide a better view of its performance.

A number of areas of improvement have been found throughout the thesis, both in the pipeline as well as the overall approach. Some of the most significant ones are briefly listed below:

1. **Anomaly scoring** — The scoring system currently performs poorly on inconspicuous anomalies, relies on static thresholding and is highly parametrised. The use of automated tuning can improve the system in the short term, while the inclusion of additional context beyond just the forecasting error is required to achieve major improvement.
2. **Expansion of sample weights** — Sample weights are currently used to remove eclipses and areas with missing data from the training data. This should be expanded to include existing and newly discovered anomalies.
3. **Choosing better target and covariate channels** — The thesis explored a wide variety of telemetry channels, but not all are equally relevant for anomaly detection. Instrument engineers and spacecraft operators could be consulted to select target channels and relevant covariates.
4. **Collecting a larger initial set of anomalies** — Improvements to the scoring system and sample weights benefit from a larger collection of initial anomalies. Those retrieved from anomaly reports in this thesis only cover a small number of events within XMM data and should be supplemented with other sources. Alternatively, an unsupervised anomaly detection pass can be used to collect an initial set of anomalies.

Although not flawless, the implemented approach proved capable of finding anomalies in XMM telemetry data. A refined approach that can be used to proceed towards an XMM-Newton anomaly benchmark has been created, incorporating solutions for the flaws discussed above and others. Additionally, considerable progress has been made in the development of software tools which can be adopted for use in anomaly detection by the XMM-Newton team and for the continued development of a benchmark. Tools built for pre-processing raw telemetry data as well as detection post-processing have a particularly high level of maturity.

Detecting an anomaly is only the first step in a longer process, and subsequent investigations may reveal a cause and lead to actions for a resolution. A problem with black-box machine learning models is that they do not provide insights into their decision-making. The thesis briefly explored how explainable artificial intelligence methods can be used to better understand an anomaly and its origins. Time constraints meant this area could not be explored fully, but the relevant background and intermediate findings focussed on the SHapley Additive exPlanations (SHAP) method are included in the report to facilitate future research.

Contents

List of Figures

List of Tables

List of Abbreviations

AE	Auto Encoder
AI	Artificial Intelligence
ARES	Analysis and Reporting System
ARTS	Anomaly Report Tracking System
ATAS	ARES Trend Analysis System
CCD	Charged Coupling Device
CNES	(French) National Centre for Space Studies
EPIC	European Photon Imaging Camera
ESA	European Space Agency
ESA-ADB	European Space Agency Benchmark for Anomaly Detection in Satellite Telemetry
ESAC	European Space Astronomy Centre
ESOC	European Space Operations Centre
Explainable AI	Explainable Artificial Intelligence
FDIR	Fault Detection, Isolation, and Recovery
GMM	Gaussian Mixture Models
HBOS	Histogram-Based Outlier Score
IOPS	Related to instrument anomaly reports
JAXA	Japan Aerospace Exploration Agency
JPL	Jet Propulsion Laboratory
LCL	Latching Current Limiter
LIME	Local Interpretable Model-agnostic Explanations
LSTM	Long short-term memory
MAE	Mean Absolute Error
ML	Machine Learning
MOS	Metal–Oxide–Semiconductor
MSE	Mean Square Error
MUST	Mission Utility and Support Tools
NASA	(American) National Aeronautics and Space Administration
OM	Optical Monitor
PN	Not an abbreviation, refers to EPIC-PN one of the imaging sensors aboard XMM-Newton
R2 score	coefficient of determination
RGS	Reflection Grating Spectrometer
ROC AUC	Area Under the Receiver Operating Characteristic Curve
SaaS	Software as a Service
SC	Related to platform or SpaceCraft anomaly reports
SHAP	SHapley Additive exPlanations
SMAPE	Symmetric Mean Absolute Percentage Error
SPK	SPICE kernels
TSAD	Time Series Anomaly Detection
VAE	Variational Auto Encoder
XMM	XMM-Newton

1. Introduction

Satellites are complex interconnected systems that, upon launch, are beyond the reach of hardware-based monitoring. Spacecraft often feature Fault Detection, Isolation, and Recovery (FDIR) systems to catch and resolve common problems. However, the complexity of modern spacecraft means that not all scenarios are accounted for. Additionally, the harsh temperature and radiation environment in space as well as software and hardware issues, inevitably lead to unexpected and unwanted changes in behaviour. Failure to catch and resolve these deviations may compromise the mission, or in the worst case, lead to catastrophic failure [1]. A prime example of this is the 2016 Hitomi breakup event [2], where an uncaught issue in the attitude determination system led to a complete loss. Such failures are costly, both economically and in terms of scientific return. To prevent them, satellites transmit housekeeping telemetry data from sometimes thousands of onboard sensors to ground stations on Earth, where it is continuously monitored for anomalies.

Monitoring telemetry for anomalies is an expensive and time-consuming endeavour, requiring specialised ground operators with expert knowledge on the spacecraft and its systems. To aid operators, ground segments commonly employ automatic monitoring systems, which raise out-of-limits warnings when telemetry signals go beyond pre-defined thresholds. Operators also regularly perform trend analyses to monitor the health of the spacecraft over time [3][4]. Some satellite operations setups are supplemented by statistical pattern-matching methods to find recurring anomalies [5] [6]. Although these methods are usually sufficient to keep a spacecraft safe, they require a significant amount of expert resources, cannot provide complete coverage over all telemetry signals, and may fail to catch anomalies that remain within statistical boundaries or have a temporal component [5]. With the number of operational satellites and their complexity expected to increase in the coming years, improved and automated anomaly detection becomes a necessity.

The XMM-Newton (XMM) space telescope, operated by the European Space Agency (ESA), is dedicated to the study of high-energy cosmic X-ray sources. Although more than 25 years old, the satellite still has considerable significance to the astrophysics field, being one of the most powerful X-ray telescopes ever built. Over 8000 publications have been made using XMM and 400 are published each year [7]. As a result, the mission has received a number of extensions well beyond its 10 year design life, with the potential to operate into the mid-2030s when it is planned to be replaced [8][9]. Until then, the XMM science operations team at European Space Astronomy Centre (ESAC) are tasked with keeping its instruments healthy and have been looking into new methods to discover anomalies and anomalous trends in telemetry of the ageing satellite. The ground segment of XMM was designed with continuous monitoring and command in mind but has been significantly automated over the years. Most recently, the mission operations team at European Space Operations Centre (ESOC) has automated XMM flight dynamics planning using Machine Learning (ML) [10], as part of ESA's *A2I roadmap* [11].

In recent years, several space actors, including ESA [11] [12], CNES [4] and NASA [13] have been researching the application of machine learning in spacecraft anomaly detection. Machine learning algorithms are already used for Time Series Anomaly Detection (TSAD) problems because of their ability to learn complex patterns in large collections of multidimensional data. Consulting literature reveals a myriad of papers proposing novel methods for TSAD problems, ranging from simple clustering methods to large deep learning models [14]. Researchers rely on publicly available datasets to benchmark the performance of their TSAD algorithms. These benchmark datasets as well as the metrics commonly used to evaluate algorithms have previously gained considerable criticism [15] [16] [17], enabling significant progress towards improved methods in recent years.

Quality public benchmarks related to spacecraft telemetry remain rare, which hinders the adoption of ML-based TSAD in spacecraft operations. Recent work by Kotowski et al. (2024) [18] introduced the European Space Agency Benchmark for Anomaly Detection in Satellite Telemetry (ESA-ADB), which contains a combined 17.5 years of anonymised telemetry data from two ESA missions and accounts for the common flaws affecting earlier work. Although this is a good start, more large datasets are required for an unbiased evaluation of TSAD algorithms in spacecraft telemetry. Additionally, the anonymised nature of ESA-ADB's telemetry prevents researchers from applying physics-informed machine learning and anomaly detection.

Once detected, anomalies are investigated to find a cause and to develop a solution. A common issue with black-box machine learning models is that they provide detections without insights into their decision-making. The explainable AI research field is focussed on understanding a machine learning model's reasoning behind their predictions and decisions. The combination of explainable AI and anomalies in spacecraft telemetry is relatively unexplored but has potential to help operators find the root-cause of an anomaly faster [19].

As part of the ESA-wide push towards the adoption of modern machine learning tools, the XMM-Newton team has recently developed Python interfaces for the retrieval and analysis of XMM-Newton instrument telemetry data, making up to 25 years of telemetry data accessible to ML-based TSAD algorithms. The large volume of telemetry data makes XMM an interesting target for the application of ML-based anomaly detection and a potential source for a new anomaly benchmark. In turn, the development of ML-based TSAD techniques could aid the XMM-Newton during continued operations of the mission. In collaboration with The XMM-Newton team and Data Science Section at ESAC, this thesis explores the application of ML-Based anomaly detection on XMM telemetry data. The project makes extensive use of the ESA Datalabs platform, which is used to access the data and enables software to be developed and executed on ESA computing infrastructure.

1.1. Research objectives and questions

Using the background and motivations of the previous section, the following research objectives are derived:

OBJ1 — Find anomalies in XMM-Newton telemetry data by applying machine learning-based anomaly detection techniques.

OBJ2 — Facilitate research into spacecraft telemetry anomaly detection by initiating the construction of a dataset of anomalies in XMM-Newton instrument telemetry data.

OBJ3 — Enable the use of machine learning-based anomaly detection techniques in XMM-Newtonn telemetry by developing a set of software tools for processing and detecting anomalies in raw XMM telemetry data.

These objectives have significant overlap. A toolset can be made to encompass the processing of telemetry data, the detection of anomalies in said data and the subsequent compilation of a dataset. Additionally, any anomalies found can be used in an eventual benchmark dataset. The choice was made to construct the toolset in the form of a data pipeline. In such systems, input data is successively passed through a series of processes which refine, manipulate, visualise and extract information using the input data. For this thesis, anomalies are retrieved from raw telemetry data through an *Anomaly Pipeline*. The choice to construct this anomaly pipeline was made for convenience, personal development and to facilitate continuation of the project beyond the thesis.

A number of topics relevant to TSAD and TSAD in spacecraft telemetry are not covered in this thesis. To start, the emerging field of onboard fault and anomaly detection is not covered in this thesis. XMM has no capacity to perform any machine learning onboard and limiting TSAD algorithms to the limited computational power of edge computing devices is not useful to the discovery of anomalies. The thesis also does not deal with streaming anomaly detection, where anomaly detection is applied live as data comes in. Instead the focus is to look at (a part of) the historical telemetry data. Finally, the purpose of this thesis is not to develop an optimal or state-of-the-art TSAD approach. The contribution of new datasets is of greater scientific significance.

It is important to note that a complete benchmark dataset, similar to ESA-ADB was not the expected end-result as the required manpower and time required to produce such a result is well beyond the scope of a master's thesis.

To tackle the research objectives, the following research questions and sub-questions have been formulated:

RQ1 — What is a suitable approach to construct a dataset of anomalies in XMM-Newton instrument telemetry data using ML-based TSAD techniques?

- **RQ1S1 —** What are the unique or notable characteristics of XMM instrument telemetry data and how do these affect the anomaly detection methodology?
- **RQ1S2 —** Which preprocessing steps are required to transform unprocessed XMM Telemetry data into a format that is digestible by machine learning anomaly detection techniques?
- **RQ1S3 —** What format and data structure should a dataset of anomalies have to be used as a benchmark?
- **RQ1S4 —** What anomalies can be found using existing ML-based TSAD techniques?
- **RQ1S5 —** How do detected anomalies compare to existing anomalies reported by operators and instrument engineers?
- **RQ1S6 —** What are the differences in approach compared to ESA-ADB?
- **RQ1S7 —** What are the limitations and flaws of the approach used and what improvements can be made?

RQ2 — How can explainable AI methods be applied to understand a detected anomaly and its origins?

- **RQ2S1 —** Which types of explainable AI methods can be applied to TSAD?

RQ2S2 — Which temporal and inter-channel dependencies can be uncovered when analysing XMM-Newton anomalies with Explainable AI?

Here **RQ1** is seen as the primary research question as Explainable AI can only be applied once anomalies are found. Unfortunately, due to time constraints, the development of tools related to **RQ2** could not be completed and some of the questions are left unanswered. The unfinished methodology is still presented to facilitate future work.

1.2. Report layout

The thesis report is divided into eight chapters and features one appendix. Chapter 2 provides essential background on the XMM-Newton mission, time series anomaly detection, explainable AI and the proprietary ESA tools used to perform the thesis. Next, Chapter 3 explores the instrument telemetry data, investigating its unique properties and their consequences for the methodology. Chapter 4 introduces the resources, tools and approach used to detect anomalies by means of an anomaly detection pipeline. An overview of the planning used to perform the thesis is also provided here. The implementation of the anomaly pipeline which takes raw anomaly data as input and provides anomaly detections as output is detailed in Chapter 5. The unfinished explainable AI implementation is also presented there. Chapter 6 presents and analyses the resulting anomaly detections with additional context provided in Appendix A. In Chapter 7, the pipeline and its results are discussed and recommendations are provided to proceed towards a new anomaly benchmark. Finally, Chapter 8 revisits the research questions and provides conclusions on the thesis work.

2. Background

The work done for this thesis spans a number of research fields. This chapter provides an introduction to several topics relevant to the completion of the project. Section 2.1 provides background on the XMM-Newton mission and its instruments, environment and operations. Next, Section 2.2 introduces the topic of time series anomaly detection. It covers the field in general, including existing algorithms, benchmarks and trends with a spotlight on anomaly detection in the context of spacecraft telemetry. Section 2.3 introduces the topic of explainable AI. Finally, Section 2.4 introduces the ESA-proprietary tools that will be used to perform the thesis work.

2.1. XMM-Newton

A good understanding of a spacecraft is vital when performing analysis of its telemetry. This section provides an overview of the XMM-Newton mission. Section 2.1.1 introduces the mission including its relevance in the field of X-ray astrophysics. Next, Section 2.1.2 provides an overview of the design of the spacecraft as well as its instruments. The orbit and environment of the spacecraft are briefly described in Section 2.1.3. Finally, Section 2.1.4 presents the operations of XMM-Newton.

2.1.1. The mission and its relevance

Cosmic X-ray sources can only be observed from space due to absorption in Earth's atmosphere [20][21]. The X-ray Multi-mirror Mission (XMM-Newton), (shown in Figure 2.1), is a space telescope operated by ESA that is used to study high-energy cosmic X-ray sources, such as black holes, galaxies, and pulsars [22][23][24]. XMM-Newton, often abbreviated to XMM and previously called the High Throughput X-ray Spectroscopy Mission, is a cornerstone mission of the ESA Horizon 2000 programme. Proposed in the 1980s and launched in December 1999, XMM was designed for 10 years of operations with an initial mission of 2 years, but has since had 8 extensions, with a current extension up to December 2026 and an indicative extension up to 2029 [8][9]. Several life-extending measures have been implemented that allow operations into the 2030s, when hydrazine reserves of the satellite will eventually be depleted [8][25] [26].

Figure 2.1: An artist rendition of XMM-Newton in orbit. Image by ESA / D. Ducros, retrieved from the ESA website [27].

After 25 years of operation, the spacecraft remains in good health and scientifically relevant, being among the most powerful X-ray observatories ever built [23]. Over 8000 publications have been made using XMM and 400 more are published each year [7]. The total number of observation hours requested exceed available hours by a factor of six [24]. Compared to other operational X-ray telescopes such as Chandra [28], launched by NASA in 1999, and XRISM [29], launched by JAXA in 2023, XMM has a high sensitivity and field of view, making it particularly capable at surveying and studying faint and extended (of large angular size, e.g. galaxies) X-ray sources, while allowing simultaneous observation in the visible spectrum thanks to the auxiliary Optical Monitor instrument [23][24].

The complete capabilities of XMM-Newton will only be replaced and exceeded by the planned Athena/NewAthena (Advanced Telescope for High Energy Astrophysics) mission which will launch no earlier than 2037 [30] [31]. XMM's most powerful counterpart, Chandra, is potentially under threat of premature cancellation due to cuts in NASA's budget [32]. Ceasing the operations of either mission would reduce our ability to perform

high-energy and time-domain astrophysics, a failure of both would prove catastrophic for the field, creating a potential multi-year gap in measurements. Thus, continued nominal operation of the ageing XMM-Newton is crucial.

2.1.2. The spacecraft and its payload

A diagram of XMM-Newton is shown in Figure 2.2. The spacecraft has a length of 10.8 meters and hosts two solar arrays spanning 16 meters in total. The spacecraft can be split into three sections [33]:

1. **Service module and Mirror Support Platform** (Shown at the bottom of Figure 2.2) — The service module hosts the spacecraft's main bus subsystems, as well as the Mirror Support Platform, which hosts three X-ray Telescopes/ Mirror Assemblies and the Optical Monitor (OM) scientific instrument.
2. **Focal Plane Assembly** (Shown at the top of Figure 2.2) — The focal plane assembly hosts the remaining scientific instruments: European Photon Imaging Camera (EPIC) and Reflection Grating Spectrometer (RGS). The EPIC instrument contains two types of sensors: MOS (Metal–Oxide–Semiconductor) and PN (not an abbreviation) which are explained further below.
3. **Telescope Tube** — A 7 meter carbon tube connecting the two payload sections, providing the focal length of the spacecraft.

The service module contains the attitude control system. Absolute pointing reference is provided by two star sensors and a fine sun sensor. Gyroscopes are used as backup during eclipses and when the star trackers are blinded. Fine attitude control is provided by four reaction wheels [25]. Finally, 8 hydrazine reaction control thrusters are used for orbit corrections, attitude control and reaction wheel desaturation. The spacecraft has a pointing accuracy of 1 arcsecond over 2 minutes [24][34].

The three X-ray telescopes each containing 58 mirrors are the namesake for XMM. They focus incoming X-rays into the instrument sensors and eliminate stray light. Two telescopes are capped with reflection grating assemblies which split X-rays into the two EPIC-MOS sensors and the two RGS sensors. The final mirror module has no gratings, directing all light to EPIC-PN [35].

XMM has three primary instruments consisting of a total of six independently operated camera sensors:

- **European Photon Imaging Camera** (EPIC) — EPIC is used for high-throughput, non-dispersive spectroscopy. It consists of three sensors, EPIC-PN (shortened to PN), EPIC-MOS1 and EPIC-MOS2 (shortened to MOS1 and MOS2). The terms ”PN” and ”MOS” refer to the type of Charged Coupling Device (CCD) sensors used in each camera. The PN camera has a higher sensitivity and temporal resolution than the MOS cameras [36] [37].
- **Reflection Grating Spectrometer** (RGS) — RGS is used for high-resolution dispersive spectroscopy. It consists of two sensors, RGS1 and RGS2. The reflection grating assemblies mounted on the mirror assemblies deflect and disperse light into the RGS sensors [38]
- **Optical Monitor** (OM) — The final camera, OM is used for optical and UV-range imaging. Unlike the other instruments, OM is a self contained Ritchey–Chrétien telescope embedded in the mirror support platform. [39].

All imaging sensors are cooled passively using radiators. EPIC also contains an auxiliary instrument, the EPIC Radiation Monitor, which measures XMM's radiation environment and supplies background radiation information to the EPIC cameras.

Each instrument has its own operating modes which significantly alter both instrument behaviour and telemetry appearance. Due to the time limited scope of the thesis, most of the project's focus was put on PN and OM which require further explanation. Additional background on MOS and RGS are omitted from this thesis report and can be found in the works by Turner et al. (2001) [36] and den Herder et al. (2001) [38] respectively.

PN's imaging sensor consists of twelve CCDs split into 4 quadrants, as illustrated in Figure 2.3a. The temperature is measured for by the thermal subsystem for the whole CCD and also at each quadrant independently. These CCDs are used differently in each operating mode. Table 2.1 lists the operating modes for PN and provides a brief description for the most important modes. Figure 2.3b shows how X-rays are readout depending on the operating mode. Before reaching imaging sensor, X-rays pass through one of six filters on a filter wheel which manipulates the incoming radiation. The filters are: open, closed, thin1, thin2, medium, and thick [37].

OM has six main operating modes, shown in Table 2.2. The filter wheel of OM contains 6 filters and enables imaging in different spectral ranges of the optical and UV bands. OM has four heaters which enable control of its secondary mirror for fine-tuned focussing [39].

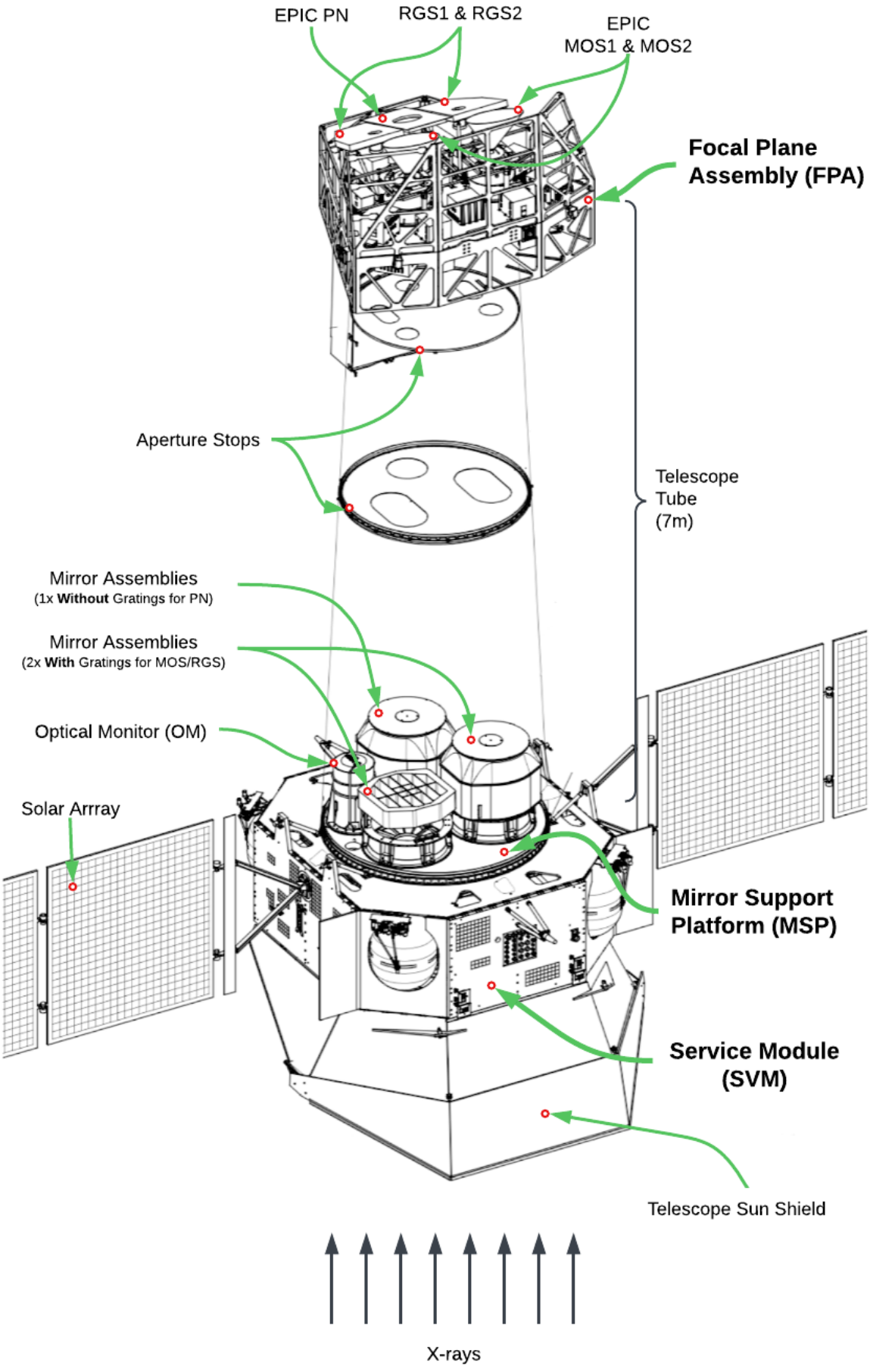


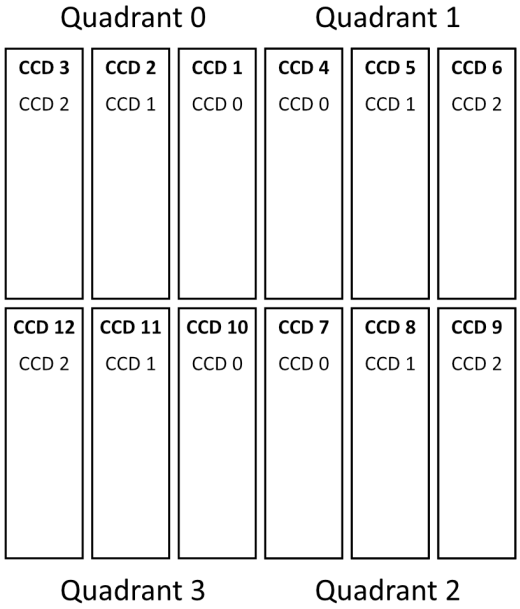
Figure 2.2: Diagram of XMM-Newton showing its most important components and instruments. Major sections are highlighted in bold. Diagram retrieved from Schartel et al. (2024) [24] and modified to include annotations for the components and instruments (with permission from the XMM team). Note that the annotations for EPIC and RGS point to the radiators at the back of the instruments.

Table 2.1: Operating modes for PN. Science modes are shown in blue. Standby modes are shown in yellow. The remaining modes are rarely used.

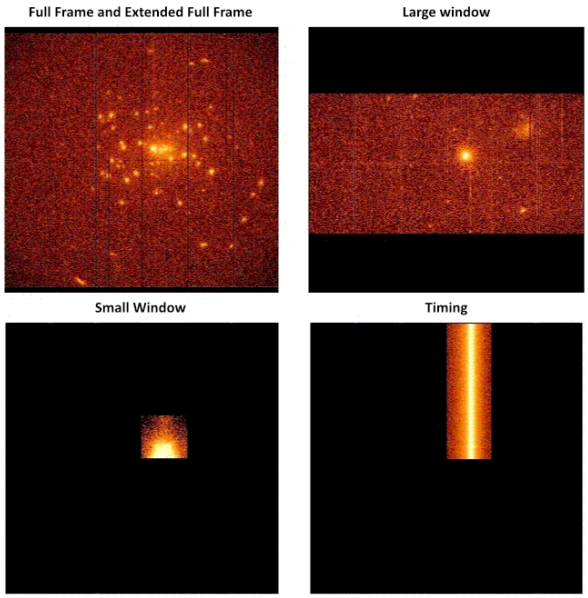
#	Mode	Description
0	FULL FRAME	Most common science mode.
1	EXTENDED FULL FRAME	For extended x-ray sources.
2	SMALL WINDOW	For very bright targets, uses only a portion of CCD 4.
3	LARGE WINDOW	For bright targets, uses a portion of all CCDs.
4	TIMING	For bright, time variable targets. Reads CCD 4 row by row in a single dimension at a high temporal resolution.
5	BURST	Similar to timing with an even faster readout at a lower quality.
6	UNKNOWN	Applied to corrupted telemetry or when the true mode is not known.
7	SAFE STANDBY	Applied during eclipses and major incidents.
8	IDLE	Applied during low-risk downtime such as periapse passes.
9	OFFSET/NOISE	
10	DIAGNOSTIC	
11	EXTRAHEATING	
12	INFLIGHTTEST	

Table 2.2: Operating modes for OM.

#	Mode	Description
0	INITIAL	
1	SAFE	Used for eclipses and major incidents
2	IDLE	
3	SCIENCE	Science mode
4	ENGINEERING	
5	INTER SAFE	Used for periapse passes



(a) Diagram showing the twelve individual CCDs of PN. Overall CCD numbering is shown in bold, non-bold numbers show the index of a CCD within a quadrant. Figure is own work to showcase multiple numbering schemes.



(b) A comparison of the readouts for PN's science modes, highlighting how the CCDs are utilised. Burst mode operates similarly to timing mode. Retrieved from the XMM-Newton user's handbook [40].

Figure 2.3: Overview of PN CCDs and modes.

2.1.3. Orbit and spacecraft environment

XMM is placed in a highly elliptical orbit with a 48 hour period, shown in Figure 2.4. Perturbations from the Sun, Earth and Moon significantly alter the spacecraft's orbit over time. XMM's orbital period is critical to its operation and is maintained without using additional fuel by utilising strategically timed momentum exchanges

with the spacecraft's reaction wheels to maintain the orbit's semi-major axis. Orbit maintenance without the use of fuel reserve is crucial to the longevity of the mission [26].

To protect the spacecraft's instruments from Earth's radiation belts, no observations are performed when passing near the pericentre. The spacecraft's changing orbital plane affects the amount of available observation time which is approximately 36 hours as of 2025, down from almost 40 hours at launch. The spacecraft orientation with respect to the Sun is also tightly controlled to ensure thermal stability, to fulfil power requirements and to keep the instruments safe from direct exposure. There are similar avoidance angles in place for the Earth, Moon and other bright planets [24].

XMM's orbit makes it susceptible to eclipses when it passes behind the Moon and Earth which require special operations. During an "eclipse season", usually lasting a few weeks, the spacecraft regularly experiences Earth-induced eclipses during each periapse passing. The number of eclipses in a season, their duration as well as the timing of the season within a year varies due to XMM's changing orbit. Elsewhere, there are rare singular occurrences where the Moon blocks the spacecraft [41].

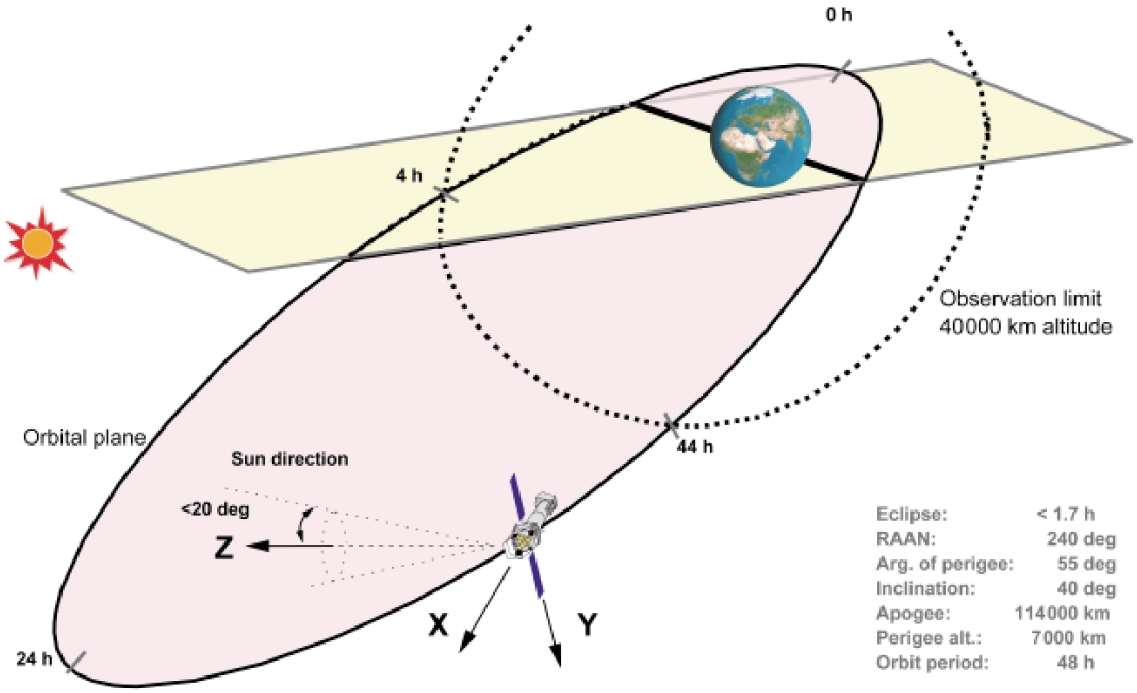


Figure 2.4: Diagram of XMM's orbit, highlighting its eccentricity and orientation towards the Sun. Image represents the situation at the start of the mission. Retrieved from Barre et al. (1999) [42].

2.1.4. Operations

XMM is designed on an outdated operations concept which requires continuous contact with the ground segment for command and downlink. It depends on a 48 hour orbit to maintain a line of sight with 4 ground stations in Argentina, Australia, Chile and French-Guyanna. Notably, XMM has no onboard mission timeline or data storage capacity. This means all commands planned on the ground are transmitted directly to the spacecraft. Data must be transmitted continuously and any loss in signal results in loss of both science and housekeeping data. From mid-2014 to 2020, unfortunate geometry of the spacecraft's orbit left short gaps its ground connection at each periapse pass. The closure of a ground station in late 2015 caused additional data loss [26].

The spacecraft has very little autonomy onboard and limited memory capacity has prevented the introduction of significant onboard automation. Most of the automations onboard are related to the monitoring of the spacecraft and critical safety measures such as heater control and recovery procedures during loss of signal. Instead, engineers have focussed on automating the spacecraft's operation from the ground as much as possible. Initially, mission controllers had to manually upload command procedures to the spacecraft. Since 2008, many of these procedures have slowly been automated from the ground to improve safety, improve efficiency, and reduce cost. Major changes include the automatic handling of eclipse procedures and an automatic Fault Detection, Isolation, and Recovery (FDIR) system that can handle known anomalies [41][43]. In total, these automation efforts have reduced the number of manually uploaded procedures by almost 80% [44].

2.2. Spacecraft anomaly detection

This section covers the topic of anomaly detection with a broad introduction in Section 2.2.1 followed by an overview of algorithms in Section 2.2.2 and benchmarks in Section 2.2.3 and finally a focus on anomaly detection on spacecraft telemetry in Section 2.2.4.

2.2.1. Time series anomaly detection

Data generated in real-world environments often contain outliers or anomalies, data points which do not conform to the expected behaviour of the system that generates the data. Depending on the context of the data, it may be practical (or critical) to detect and understand the cause of anomalies. Anomaly detection is commonly applied in fields such as cybersecurity (e.g. detecting anomalous computer network traffic), fraud detection (e.g. fraudulent financial transactions) and bioinformatics (e.g. detecting abnormalities in medical data). For each data point, anomaly detection algorithms present their output in one of two forms:

1. An anomaly score, which provides a numerical quantification of how anomalous a point is.
2. A binary label, which describes a point as anomalous or not anomalous.

Anomaly scores can be converted to anomaly scores by means of thresholding. Time Series Anomaly Detection (TSAD) is the application of anomaly detection on time series data which contain data points that are measured over time, either at regular or irregular time intervals. Individual points in time (point anomalies) or continuous subsets of time (subsequence anomalies) may then be considered as anomalous events [45]. Figure 2.5 shows example anomalies from datasets of various disciplines.

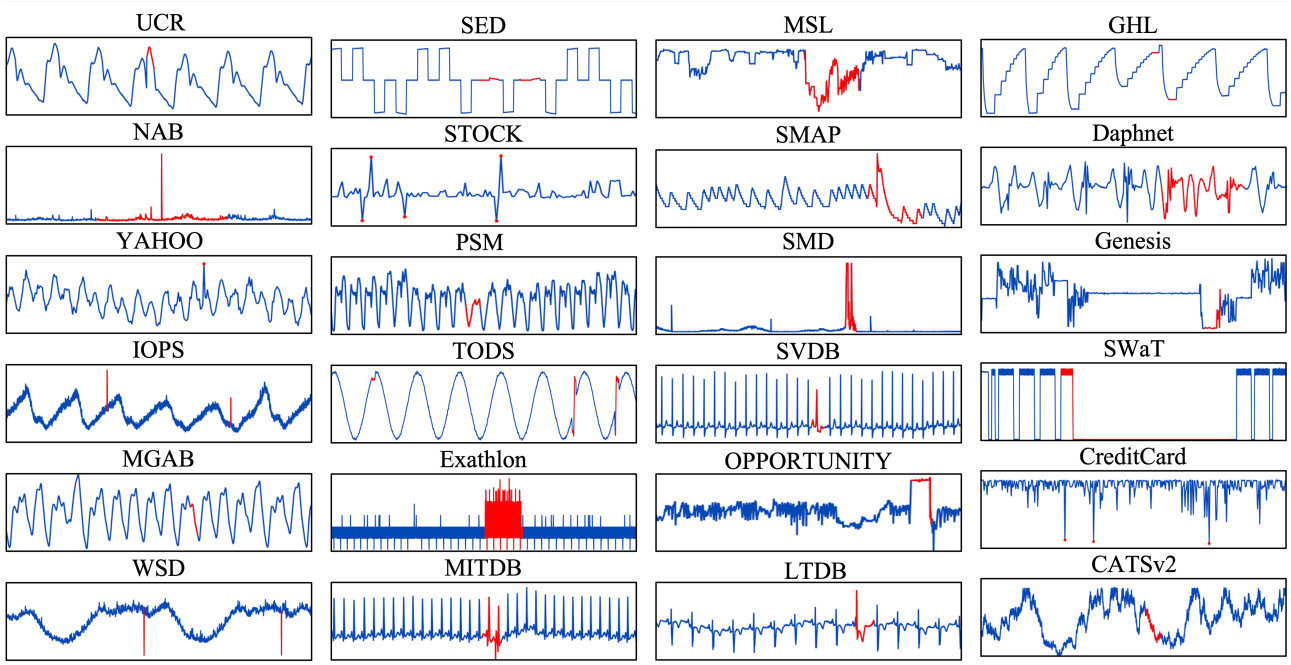


Figure 2.5: Annotated nomalies (in red) for various TSAD datasets. Annotations are made by the original benchmark authors. Retrieved from the TSB-AD benchmark repository [46]

The field of TSAD has been studied for decades, resulting in a myriad of algorithms with various levels of complexity. The simplest methods involve basic thresholding like the out-of-limits systems commonly used for spacecraft telemetry. Methods like moving average models and Autoregressive Integrated Moving Average (ARIMA) models rely on statistical methods like autocorrelations, and moving averages [45]. Due to success in other fields, much of recent TSAD research involves machine learning or deep learning. Machine learning is a broad field within data science that encompasses algorithms that can learn desirable patterns from input data without explicit instruction and can apply these learned behaviours to unseen data. Deep learning is a subset of machine learning that utilises a bio-inspired neural network architecture, replicating the function of neurons. The border between statistical methods and machine learning is artificial and blurry as many machine learning techniques are rooted in statistical tradition. Application of the machine learning label is often based on the history of algorithm and the complexity of its learning mechanism.

2.2.2. Time series anomaly detection algorithms

Varying taxonomy is used in literature to categorise the algorithms, those most useful to the project are discussed below:

Direct detection vs window-based detection vs pattern-based detection

In window-based detection, time series are split into multiple (overlapping or non-overlapping) windows, anomalies are then scored per window. In direct detection, each data point is scored individually. Finally, pattern-based detection is similar to window-based where the time series is cut into subsequences of the same pattern and scored per subsequence [47]. Direct detection is most commonly found in literature.

Univariate vs multivariate methods

A system generating time series data may contain multiple sources of data that are interlinked. As an example, satellites may contain multiple sensors that record different parts of the spacecraft at the same time. Each sensor can be said to record a variable. Algorithms that can handle multiple variables concurrently are called multivariate. Those that can handle only a single variable are called univariate. Because the term variable may be ambiguous depending on the context, an alternative term channel is used henceforth. Other terms are also used: feature (machine learning field) and parameter (satellite operations field). The terms univariate and multivariate may also be used to describe the dataset.

A multivariate dataset may distinguish between target channels (those that are targetted for anomaly detection) and covariate channels (which can be used by algorithms to retrieve additional context).

Supervised vs semi-supervised vs unsupervised learning methods

TSAD as well as machine learning methods are commonly distinguished by their level of supervision. Schmidl et al. (2022) [14] provide the following definitions in the context of TSAD:

1. "**Unsupervised** algorithms separate anomalous points from the normal part of the time series without prior knowledge (no explicit training step is required)"
2. "**Supervised** algorithms model normal and abnormal behaviour in the time series and require a training step before they can be employed on a new time series. All points of the training time series must be marked as either normal (usually 0) or anomalous (usually 1). These algorithms learn to distinguish between the normal and the anomalous behaviour of the training time series. Given an unseen test time series, the algorithms can, then, mark the anomalous subsequences that match their internal representation of anomalous behaviour."
3. "**Semi-supervised** algorithms try to learn only the normal behaviour of a training time series. This means that they should be trained on normal time series to build a model of the normal behaviour. When applied to a test time series, all subsequences that do not conform to the normal behaviour are marked as anomalous."

Readers with previous experience in machine learning may be familiar with a different definition of semi-supervised learning. In other fields semi-supervised learning involves the use a partially labelled dataset. In that sense, the definition above sets all training data to be singularly labelled as normal behaviour. It should also be noted that supervised methods are rather unpopular for TSAD because of their limited ability in finding new anomalies [14][45].

Both supervised and semi-supervised models require a separeted training and test set to avoid data leaks. An optional validation set is often also used to aid in model selection and to prevent over-fitting.

Algorithm families

TSAD algorithms can be grouped by their method of marking subsequences as anomalous. In the review by Schmidl et al. (2022) [14], which surveyed 158 algorithms and evaluated 71 of them, algorithms are grouped in the following families:

1. **Forecasting Methods** — Forecasting methods rely on autoregression: using past values of a time series to construct its future values (with expected normal behaviour). These forecasted values are then compared to the original values. Points with a high forecasting error may then be marked as having a high anomaly score. Algorithms in this family include the unsupervised *ARIMA* method, semi-supervised Recurrent Neural Networks (RNN) like *Telemanom* (which is based on a Long short-term memory (LSTM) architecture) [13] and other machine learning regression methods like the semi-supervised tree ensemble-based XGBoost [48].
2. **Reconstruction Methods** — Reconstruction methods encode the input data into a simplified latent space and subsequently reconstruct that data, with the expected behaviour for comparison with the input. Similar

to forecasting methods a reconstruction error is then used to produce an anomaly score. Methods in this family include Auto Encoder (AE) and Variational Auto Encoder (VAE) like *Donut* [49].

3. **Encoding Methods** — Similar to reconstruction methods, encoding methods cast the input data into a latent space but then proceed to produce an anomaly score from the latent space data directly.
4. **Distance Methods** — Distance methods compare data points or subsequences with each other and assign higher anomaly scores to those that are most dissimilar to other points. Methods in this family are often unsupervised methods like K-Means clustering and Histogram-Based Outlier Score (HBOS)[50].
5. **Distribution Methods** — Distribution methods use a generated distribution of the input data and assign a score to a point or subsequence based how it conforms to the distribution. A notable distinction with distance methods is the use of frequency.
6. **Isolation Tree Methods** — Isolation Tree Methods require a bit more background to understand. Decision trees are used to classify data points into groups based on a set of decisions resembling a trees with multiple levels of branches. A decision may be: ”channel exceeds 5, yes or no?”. In random trees, the channel and decision value are set randomly. Every classification starts at the root of the tree and ends in a leaf. Isolation forests employ a large number of random trees and for each tree calculate how many decisions are required to isolate a data point or subsequence within one leaf of a tree. Anomalous subsequences are expected to be easier to isolate and can be found by taking subsequences with a low average number of decisions. Algorithms in this family include Isolation Forest [51] and its variants.

For forecasting and reconstruction methods, anomaly scores are derived from a forecasting or reconstruction error, which is modified to retrieve an anomaly score. A simple and common solution is using the absolute error, but more advanced scoring functions can be found, applying statistical formulas and even unsupervised TSAD methods to the forecasting / reconstruction error [52][12]. Other families retrieve the anomaly score directly. Most methods involving deep learning are semi-supervised forecasting or reconstruction methods. These methods often also support the use of covariates in multivariate datasets.

Many machine learning methods, especially deep learning methods, require preprocessing. To start, almost all TSAD methods require input data to have regular time steps. Deep learning and Distance-based methods often use a euclidean distance and require scaling to prevent feature dominance (ensuring all features have equal weight). Additionally, deep learning methods rely on gradient descent for optimisation, which requires scaling to improve optimisation performance and stability[53] [54]. Finally, because many machine learning methods have a statistical foundation, supervised and semi-supervised methods perform best on data that has the same distribution across a dataset. Data that has this property is termed stationary.

2.2.3. Algorithm benchmarking

With so many TSAD algorithms and more created each year, researchers rely on publicly available benchmark datasets to compare their algorithms to existing ones. At their core, TSAD benchmarks contain a continuous length of time series data with one or multiple target channels, complemented by ground truth binary anomaly labels at each point in time. Multivariate datasets may have an anomaly label per target channel. Datasets are often split to accommodate supervised and semi-supervised models. A long stretch of time is split into two parts to create a training and test set. Some datasets include covariate channels, which may be used by compatible TSAD algorithms to provide additional context to reconstruct or target channel data. Researchers may then run their algorithms on the benchmark, comparing the resultant labels to the ground truth using various metrics.

Since 2018 the spacecraft telemetry-related MSL (Mars Science Laboratory) and SMAP (Soil Moisture Active Passive) benchmarks published by NASA [13] have been popular benchmarks used across TSAD research with over 1500 citations. Up to 2024, they were the only publicly available anomaly detection datasets related to spacecraft telemetry. Trends in spacecraft design make publication of telemetry data difficult, as this data is usually proprietary to a spacecraft’s manufacturer.

Recently, multiple benchmarks including MSL and SMAP have been subject to criticism. Wu and Keogh (2021) [15] argue that anomalies contained within these benchmarks are flawed and provide four main arguments:

F1. **Triviality** — A large portion of the anomalies found in the discussed benchmarks can be found using trivial approaches, such as out of limits thresholding.

F2. **Unrealistic Anomaly Density** — Benchmark datasets contain an unusually high percentage of anomalous data points and have many of their anomalies concentrated a few areas within the dataset.

F3. **Mislabelled Ground Truth** — The datasets contain anomalies that are mislabelled or inaccurately labelled. Resulting comparisons are thus flawed as an algorithm may mislabel subsequences or have alternative start and end times, detrimentally affecting performance in comparison metrics.

F4. **Run-to-Failure Bias** — Datasets like MSL and SMAP have anomalies placed at the end of their time series segments allowing for dummy algorithms which simply detect the end of each segment.

Wagner et al. (2023) [16] add three additional arguments:

F5. **Long anomalies** — Many TSAD algorithms, rely on a windowed approach to score data. Very long anomalies that extend beyond the window prevent these methods from accessing the normal behaviour surrounding the anomaly, hampering their functioning. The benchmarks criticised contain a large number of long anomalies.

F6. **Distributional shift** — also known as **stationarity** is a feature of a time series data meaning its behaviour remains the same across the lifetime of the dataset. The datasets are criticised for showcasing different behaviours across the their training and test sets.

F7. **Constant features** — Some datasets contain channels that remain constant across the dataset adding unnecessary noise to the models.

Although these flaws (except F2 and F3, but especially F5, F6 and F7) can be attributed to the use of real-world data and are sometimes impossible to correct, current academic consensus has marked these datasets as unusable. In addition to the datasets, the metrics used to measure the performance of a benchmarked algorithm has also received criticism [16][17]. Authors often use the ”Point-adjusted” F1 score, which marks an entire anomalous subsequence as correctly detected if even one data point in the subsequence is detected. The method may thus significantly overestimate the performance of a measured algorithm. As a result, much of the literature related to novel anomaly detection techniques is rendered unreliable.

A few benchmark surveys acknowledge and account for the criticisms discussed above:

- Schmidl et al. (2022) [14] benchmark 71 algorithms of various families and conclude that no singular algorithm or family of algorithms performs best across all benchmark datasets tested. Some families appear to work best on specific types of anomalies but all families can work well across different benchmarks. Extreme value anomalies are easiest to find while anomalies involving trends are the most difficult. Reconstruction methods in particular struggle with trend anomalies. The authors also write that current deep learning methods are not competitive compared to other types of algorithms, although this result can be seen as biased. A choice to set a low memory limit of 3 GB and a 3 hour runtime limit meant many computationally intense deep learning methods were excluded from the results. Despite these conclusions, aggregate metrics show forecasting methods slightly outperforming other multivariate algorithms while distance-based methods perform best among univariate algorithms.
- Wagner et al. (2023) [16] compare a number of deep learning based algorithms on two datasets and similarly conclude that no one algorithm performs best on either dataset. The authors use a grid search to tune each model. Both auto encoder-based reconstruction methods and predominantly LSTM based forecasting methods have promising results. Generative Adversarial Networks tested showed the worst results.
- Herrmann et al. (2024) [17] benchmark a few popular TSAD algorithms on a proprietary spacecraft telemetry dataset finding LSTM-based methods to perform best.
- Liu et al. (2024) [55] evaluate 40 algorithms on 40 datasets, tuning each algorithm. The results are presented using various evaluation metrics. Here neural network based methods were found to perform best in multivariate problems and simpler model architectures were again found to perform best. Conclusions from this work are not taken into account for this review as it was released very late into the thesis (Nov 2024). It is included here as a useful resource for future readers.

The lack of a standard benchmark metric and the aggregate presentation of results make it difficult to compare algorithms. Additionally, most methods have many adjustable parameters and the amount of tuning, whether across datasets or on individual ones, may significantly affect measured performance. Ultimately, only two core conclusions can be made: 1. No one algorithm performs best on all datasets and 2. simple algorithms may perform as good or better than more intricate counterparts.

2.2.4. Anomaly detection for spacecraft telemetry

The identification of the issues mentioned in the previous section will enable more reliable research in the coming years and researchers remain excited about the use of machine learning-based anomaly detection for spacecraft telemetry applications. ESA has noted mission operations as an important machine learning application in their A2I roadmap [11], highlighting how machine learning can be applied to anomaly detection and root-cause analysis. Additionally, several institutions have been researching the use of machine learning for on-board anomaly detection [56][57][58].

To facilitate research into anomaly detection for spacecraft-telemetry, Kotowski et al. (2024) [18] have introduced the European Space Agency Benchmark for Anomaly Detection in Satellite Telemetry (ESA-ADB), which is a collaboration between experts in spacecraft telemetry and machine learning and aims to solve the problems presented in the previous section. The authors identify a number of aspects that make satellite telemetry an especially challenging field within TSAD:

1. Telemetry has a high dimensionality (number of channels) and volume (duration / number of data points).
2. Telemetry channels are highly dependant on each other.
3. Raw telemetry data has channels with varying sampling rates, irregular acquisition times and a number of missing data points related to communication gaps.
4. Telemetry data is seasonal and evolves with operational phases, operational modes, the orbit and component degradation.
5. Telemetry data contains various types of channels, including mode channels, status flags, counters and a number of different physical measurement units.
6. Telemetry data is affected by noise and measurement errors caused by the space environment.

The work contains two benchmark datasets, each constructed from anonymised telemetry data from an ESA mission and named *Mission1* and *Mission2*. These datasets are complemented with an evaluation pipeline including metrics designed to meet the specific needs of satellite operators. The datasets feature a large amount of data (14 + 3.5 years) with less then 2% annotated as anomalies. Anomalies are found using multiple unsupervised and semi-supervised detection passes and ultimately hand-annotated per channel [59]. Each mission is noted to have unique challenges for TSAD algorithms. A number of channel subsets and time ranges are suggested to allow for a reduced challenge and intermediate testing. A third mission, *Mission3* is investigated but excluded as a benchmark dataset due to a large amount of communication gaps and a low number of anomalies.

In addition to annotating anomalies, a number of rare nominal events and communication gaps are annotated. Rare nominal events are described as atypical events that would typically be caught as anomalies by an algorithm but are seen as expected or planned from the perspective of a satellite operator. Both anomalies and rare nominal events are classified into groups described by satellite operators and have a number of categorisations to allow for a more detailed performance analysis. Additionally, the anomalies are assigned the following attributes:

- **Dimensionality** — An anomaly is multivariate if there multiple channels affected at the same time by an anomaly. They are univariate if the anomaly only affects one channel at that time.
- **Locality** — Channels have nominal minimum and maximum values. If the values of an anomalous subsequence lay outside this nominal range, the anomaly is said to be global. Otherwise, it is said to be local.
- **Length** — An anomaly is marked as a point anomaly if it consists of up to 3 data points. Longer anomalies are termed as subsequence anomalies.

Finally, eight popular TSAD algorithms were benchmarked on the two datasets. Despite the use of powerful desktop hardware (with high-performance CPUs and GPUs as well as more than 32 GB of RAM), some algorithms could only be benchmarked after some modification. A number of popular models had to be discarded completely. None of the algorithms tested on the complete dataset show remarkable results, highlighting the difficulty posed by a realistic anomaly dataset. In datasets with a reduced set of channels, algorithms achieve better results with versions of the LSTM-based Telemanom algorithm [13] and Isolation Forest [51] performing best on *Mission1* and *Mission2* respectively. Multivariate algorithms like Telemanom were found to be especially useful for dealing with covariate channels like telecommands, statuses and modes.

ESA-ADB is the most expansive spacecraft telemetry dataset available to the public, and it has been used both as a source of inspiration and comparison throughout the thesis.

2.3. Explainability in machine learning

Recent advances in computer hardware have enabled the development of increasingly complex and performant machine learning and deep learning models. However, their inherent complexity prevents a direct derivation their decision-making process. Such models, with known inputs and outputs but uninterpretable internal workings, are termed opaque or black box models. Ethical and safety concerns in adoption areas such as medicine, law and security, which require increased accountability and trust, have raised a growing interest in Explainable Artificial Intelligence (Explainable AI / XAI) [60][61]. The topic covers the concepts of *explainability*; the ability to understand the internal workings of a model and *interpretability*; the ability for humans to understand the cause

of their decision-making or for an AI to explain its decision-making to humans [60]. The latter is particularly interesting to spacecraft operations as an ideal implementation could potentially be used to explain the origin of anomalies.

Transparent machine learning algorithms, such as decision trees and logistic regressors, exist but are rarely able to achieve the same performance as their black-box counterparts [60]. Instead, research has largely been focussed on methods that introduce interpretability to existing models. A number of those solutions exist with both model-agnostic and model-specific flavours. As an example for the latter, a number of methods exist that extract neural network weights or features like attention heads to interpret their output. These methods were avoided in favour of model-agnostic methods, as the latter do not require the project to be tied to a specific model. Depending on the method, Explainable AI can provide global interpretability, which provides insights on an entire model, and local interpretability which provide insights on an individual prediction. Finally, methods can be distinguished their supported data types, such images, text or tabular data.

In the context of this thesis, which is focussed on time series data, only methods that support tabular data are considered. Additionally, as not all time steps are anomalous, a method should be able to interpret individual or segments of time steps, requiring support for local interpretability. The two most popular methods that meets these requirements are SHapley Additive exPlanations (SHAP) [62] and Local Interpretable Model-agnostic Explanations (LIME) [63]. SHAP was preferred, because it is actively maintained, allows for consistent comparison between multiple local interpretations (useful for comparing anomalous vs nominal behaviour) and also allows for global interpretation. The remainder of the section is used to explain the method and its properties.

For each individual prediction, SHAP assigns an importance value to all input features based on their contribution to the output. The resulting Shapley values are based on a game-theoretic approach to fairly split a pay-out between collaborators. SHAP incorporates previous methods like LIME and can be applied to a variety of data types, including images, tabular (including time-series) and text data. In these cases a feature may be a pixel, a covariate's value or a word in a sentence. Interpretability is then achieved by aggregating and visualising the Shapely values in various ways. Figure 2.6 shows a clear example applied to image classification.

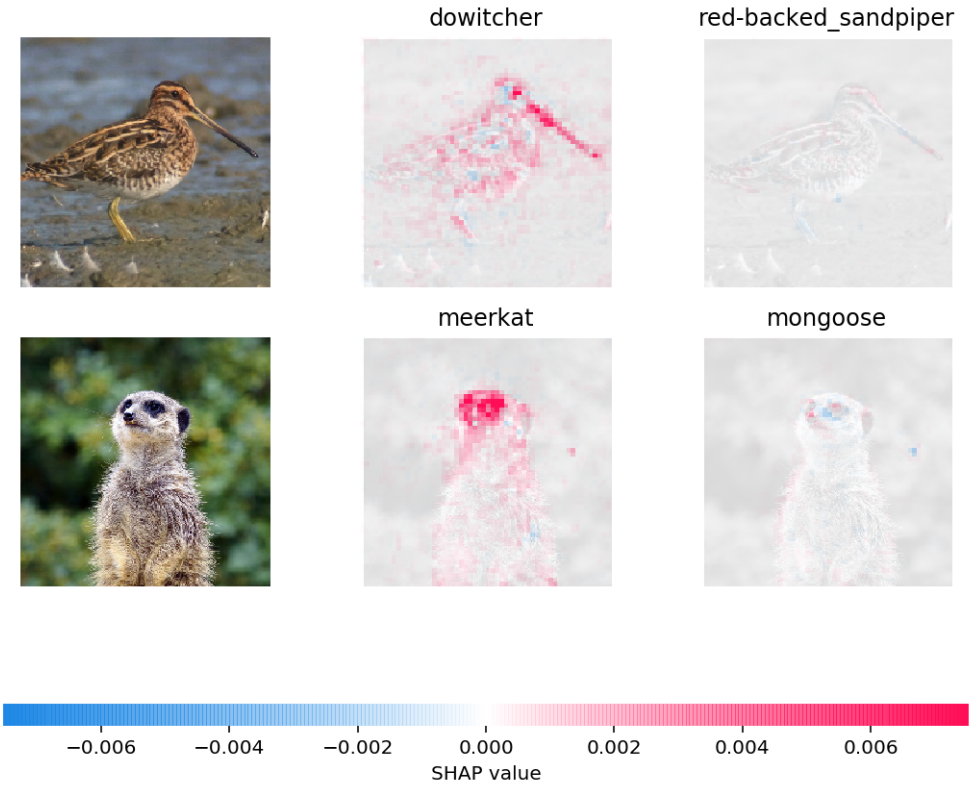


Figure 2.6: Example application of SHAP on an animal image classification dataset. Higher SHAP values denote a higher contribution of those pixels to a particular choice in animal. In this case, the long bill of the dowitcher and distinct eyes of the meerkat are strong contributors to their respective classifications. A red-backed sandpiper and mongoose would not have such body parts. Retrieved from the SHAP GitHub repository [64]

Gomez et al. (2024) [12] apply SHAP to a spacecraft-telemetry TSAD problem, utilising it to understand relationship between telemetry channels and gain insight into the origin of anomalies in telemetry data from the

Euclid space telescope. They retrieve a forecasting error by applying an XGboost model which is then modified by applying K-means clustering to retrieve an anomaly score. Finally, a second XGboost model is trained to predict the anomaly score, which is subsequently analysed using SHAP to see which covariates contribute most to a subsequence with a high anomaly score.

SHAP has a number of noteworthy properties that must be considered during implementation:

1. **Global and Local interpretability** — As mentioned previously, SHAP supports both global and local interpretability. In the context of time-series, SHAP values are calculated per covariate channel per time step. The sum of these values will then always equal the difference between the local model output and the global mean, allowing for consistent comparison and aggregation [61]. As an example, the sum across all time steps can be used to show the global importance of each covariate.
2. **A high SHAP value does not necessarily imply causation** — SHAP assigns a value based on the contribution of each covariate, but this does not imply causation [61]. In satellite telemetry, multiple channels may be affected by a single anomaly and analysis on one channel will then yield high SHAP values on other affected channels, even though the origin of the problem may lie elsewhere.
3. **Interpreting SHAP visualisations is not always straightforward** — The example shown in Figure 2.6 is quite easy to understand: positive SHAP values contribute to a likelihood of that class, while negative values argue against that classification. Other examples, such as Figure 2.7 are more difficult to interpret.
4. **Multiple SHAP explainers** — The SHAP library contains interfaces that cater to various machine learning architectures. The most notable ones are the *DeepExplainer* for neural networks, the *TreeExplainer* for decision tree-based models like XGBoost and the *KernelExplainer*, which applies linear regression internally to retrieve SHAP values for any model type. The *TreeExplainer* is generally the fastest version and is most compatible with high amounts of data while *KernelExplainer* should be avoided in such cases.

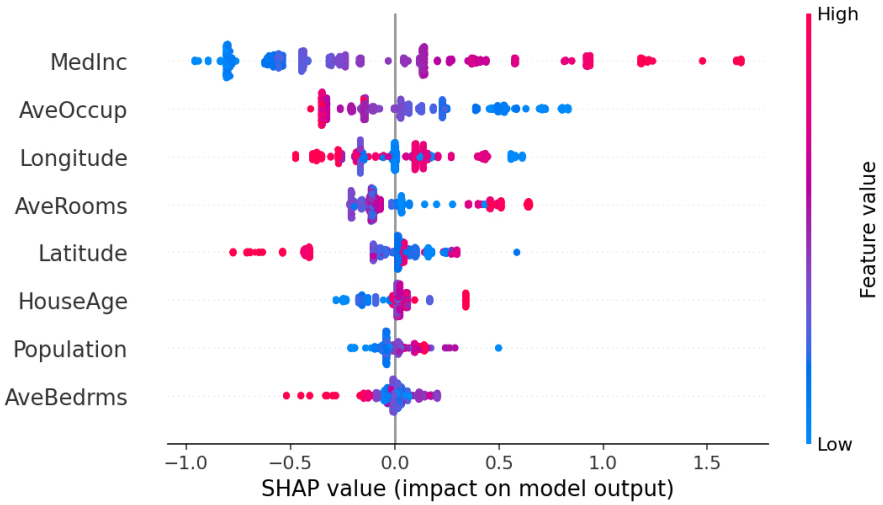


Figure 2.7: Another example application of SHAP applied to housing market regression, highlighting a more difficult case to understand. Each dot represents the median house price of a neighbourhood, with pink colours representing expensive ones. As an example, a high neighbourhood median income (MedInc) is a strong indicator for a high housing price. Other properties like the longitude are even more difficult to interpret. Retrieved from the SHAP GitHub repository [64].

2.4. ARES, ATAS and ESA Datalabs

As of early 2025, the European Space Operations Centre (ESOC) collects telemetry data from 21 active spacecraft. To store and analyse the immense volume of housekeeping data, ESOC uses the Analysis and Reporting System (ARES). The system succeeds the previously used Mission Utility and Support Tools (MUST). ARES Trend Analysis System (ATAS) is a proprietary python library created in the XMM team to retrieve data from ARES and produce automatic instrument telemetry reports and analysis. ATAS is used as a starting point for the data retrieval efforts of this thesis.

ESA Datalabs [65] (datalabs.esa.int) is an online Software as a Service (SaaS) platform developed at ESAC. The platform is accessed through a web browser and provides user with direct access to various large data caches maintained by ESA. Users can then readily exploit the data using the service's compute capacity. In the context of this thesis, Datalabs is used for all programming efforts. It is used to access ARES, develop the anomaly detection toolset and analyse results.

In the context of this thesis, Datalabs is used to access telemetry data from ARES and develop the entire toolset. An example screenshot of the website is shown in Figure 2.8

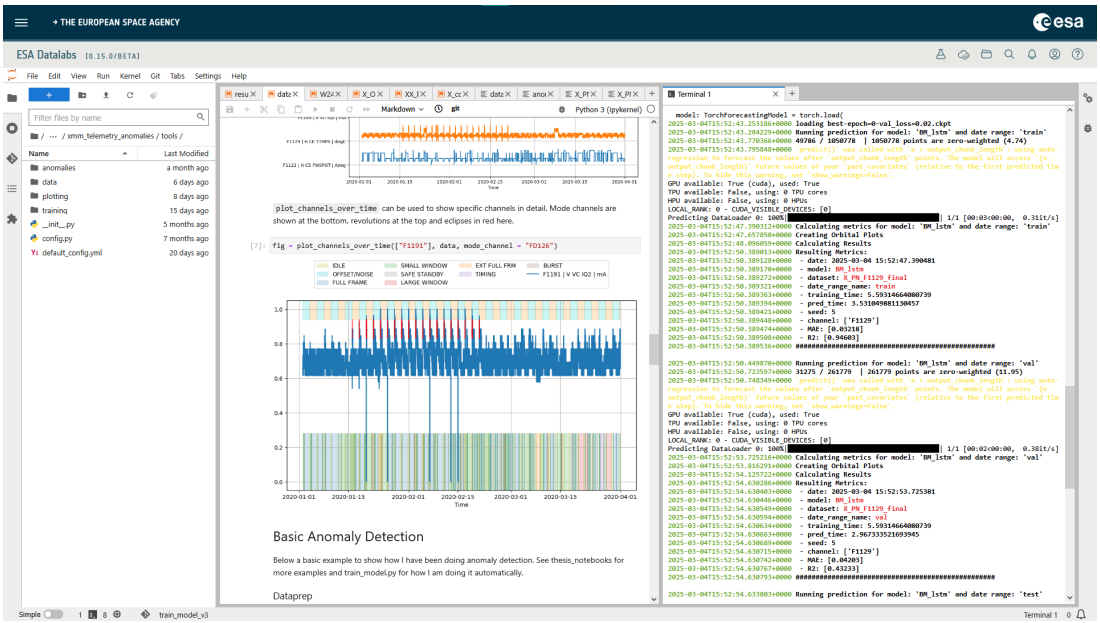


Figure 2.8: Example screenshot of the ESA Datalabs platform, showing its coding interface.

3. Data Exploration

In order to produce a methodology for detecting anomalies in XMM-Newton, a core understanding of the telemetry is required. At the start of the thesis, a few weeks were spent on exploratory data analysis. The results of that research phase as well as various later findings are summarised in this chapter and form the basis of the methodology presented next in Chapter 4. The chapter is divided into three sections. Section 3.1 presents the instrument telemetry data, focussing on OM and PN, the two instruments explored in this thesis, showing both their general characteristics as well as quirks specific to XMM. Next, Section 3.2 presents a brief overview of existing anomalies contained in reports generated by instrument engineers. Finally, Section 3.3 contextualises the results of previous sections by comparing the telemetry to ESA-ADB and summarising the effects the data has on the methodology.

The data used in this section spans the years 2014 to 2023, in line with limits discussed in Section 4.1.1.

3.1. Overview of telemetry

This section provides a broad overview of the instrument telemetry data. General characteristics, such as the sample rates and a view of the telemetry data are provided in Section 3.1.1. An important quality for the functioning of machine learning algorithms: stationarity is discussed in Section 3.1.2. The primary instrument modes, which are strongly linked to the behaviour of telemetry are presented in Section 3.1.3. Eclipses, which periodically affect XMM are shown in Section 3.1.4. Finally, Section 3.1.5 takes a closer look at the availability of various channels.

3.1.1. General characteristics

Information from thousands of telemetry channels are downlinked from XMM-Newton but only few are stored and analysed by instrument engineers at ESAC. Telemetry channels may contain a variety of information, such as sensor data (e.g. temperatures currents and voltages) as well as instrument modes, counters and status flags. A core collection of telemetry channels, commonly monitored by instrument engineers in instrument reports was retrieved from ATAS and used throughout the thesis project. Platform related telemetry, such as telecommands, are currently not stored in the system and thus not used. Telemetry channels are identified by a five symbol code: often but not always, a letter denoting the subsystem and 4 numbers (e.g. H5120). For some instruments, relevant channels from other subsystems such as thermal are also monitored. Table 3.1 provides the number of channels and their units and subsystems per instrument.

Table 3.1: Catagorisation of the various instrument channels used in instrument reports at ESAC. Square brackets [V] denote a unit, parentheses (F) denote the subsystem prefix of a telemetry channel ID, eg. F0000 for PN and T0000 for the thermal subsystem.

	PN (F)	MOS1 (E)	MOS2 (K)	RGS1 (G)	RGS2 (L)	OM (H)
Number of channels	52	36	36	177	177	46
Mode channels	2	2	2	3	3	7
Instrument-specific	50	34	34	166	166	37
Thermal Subsystem (T)	2	2	2	7	7	7
Power Subsystem (P)	0	0	0	4	4	2
Temperature [°C]	8	14	14	15	15	14
Current [mA / µA / A]	29	0	0	29	29	10
Voltage [V / mV]	4	19	19	124	124	7
Power [W]	0	0	0	1	1	0
Other units / no unit / modes	11	3	3	8	8	15

Table 3.2 lists the approximate availability and ideal sample rate of the various channels of OM and PN. XMM telemetry data is sampled and downlinked at varying rates and the ultimate acquisition times vary slightly every time. Additionally, the data contains a number of gaps of varying durations and causes which are further explored in Section 3.1.5. These properties are common to all spacecraft telemetry and have already been described in the context of ESA-ADB in Section 2.2.4. The terms ’gap in data’, ’missing data’ and ’communication gap’ are sometimes used interchangeably in this report.

The data processing required to modify the data into regular time steps and deal with gaps in data is discussed in Section 5.1.1, but is already applied to most of the data shown in this chapter to facilitate the generation of

aggregate data. The sample rates of Table 3.2 are retrieved from the raw data.

Table 3.2: Availability of the various channels of OM, PN and relevant thermal and power subsystem channels. Sample rate listed is found by measuring the most common time distance between two samples for all samples in raw data between 2014 and 2023. Percentage of data missing is calculated by collating data to 1 minute and calculating the percentage of minutes where no samples are found for data between 2014 and 2023. [*] This number is slightly misleading due to the choice to collate data at 60 seconds which is lower than the sample rate.

Instrument	Channels	Sampled every	Data missing
PN	FDxxx (modes)	5 seconds	6.9%
	F11xx, F12xx, F13xx, F14xx	7 seconds	6.7%
	F15xx, F16xx, F17xx, F18xx (CCD quadrants)	63 seconds	13% *
	T4004, T4005 (Thermal subsystem)	3 seconds	6.7%
OM	Hxxxx	10 seconds	7.1%
	Txxxx, Pxxxx (Thermal/ Power subsystem)	3 seconds	6.4% - 7.5%

Telemetry data itself has varying appearances, as shown in Figure 3.1 for OM and Figure 3.2 for PN. Some channels, especially those of OM (e.g. T0004 and H5120), show clear repeating patterns while others do not. Many of them, most notably voltages, are heavily discretised, showing few possible values across their lifetime. It can also be observed how localised channels, such as those related to the quadrants and individual CCDs of PN (e.g. F1571 and F1576) have a higher level of discretisation than channels that are more globally relevant (e.g. F1193 and F1128). Finally, some temperatures have seasonal relationships with the orbit of XMM around the Earth (e.g. T4004 and F1193) and by extension the orbit of Earth around the Sun (not visualised).

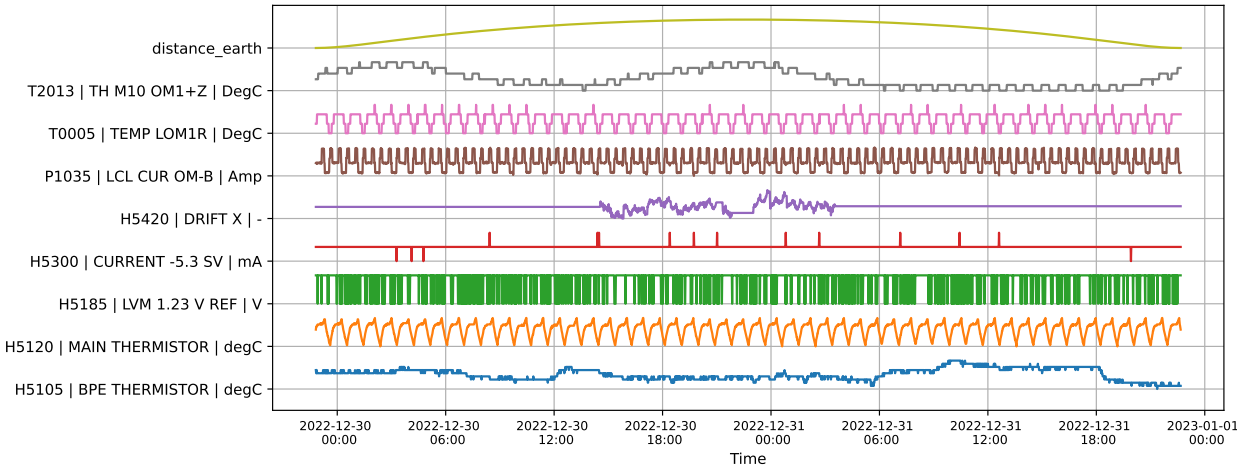


Figure 3.1: A selection of telemetry channels for OM over a single orbit, highlighting varying the discretisation, seasonalities, and varying appearances of the parameters. Values are scaled.

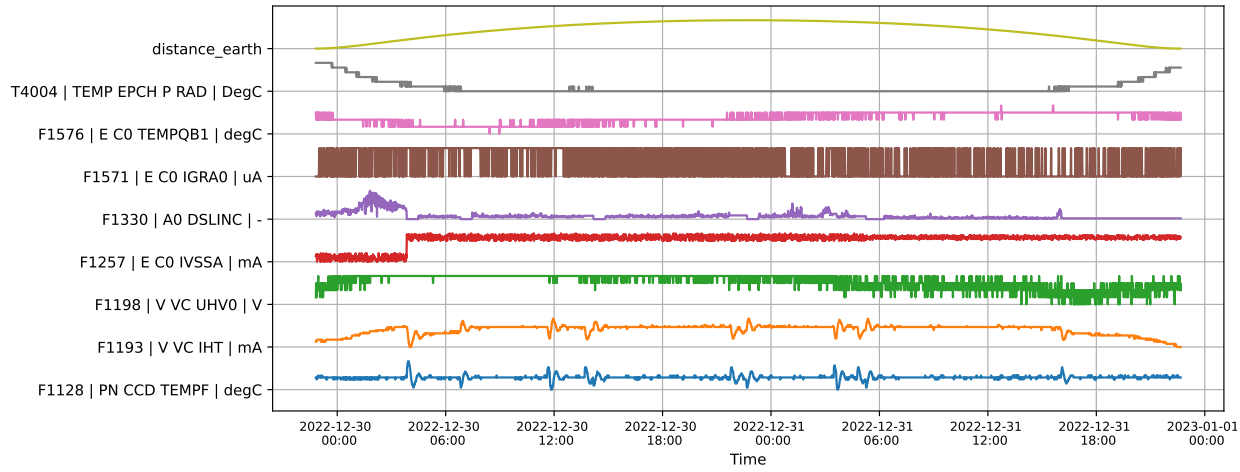


Figure 3.2: A selection of telemetry channels for PN over a single orbit, highlighting the discretisation, seasonalities, and varying appearances of the parameters. Values are scaled.

3.1.2. Stationarity

As described in Section 2.2.2, machine learning algorithms work best when presented with stationary data: data that retains the same statistical properties across its lifetime. Unfortunately the reality is far from this ideal. The nominal operational regime of a satellite is frequently updated, especially in the case of XMM which has seen a number modifications to improve its efficiency, improve automation and combat degradation. A few major examples have already been noted in Section 2.1.4. Additionally, external factors such as the orbit of the spacecraft affect how the satellite has to operate. Figure 3.3 shows how the values of various channels in OM and PN change over time by aggregating mean values each quarter. Some channels remain relatively stable, occasionally displaying small seasonalities over a year. Others display significant drift over time. The stationarity of a channel affects its ability to be predicted over long periods of time and this property is taken into account when selecting channels for analysis in subsequent sections.

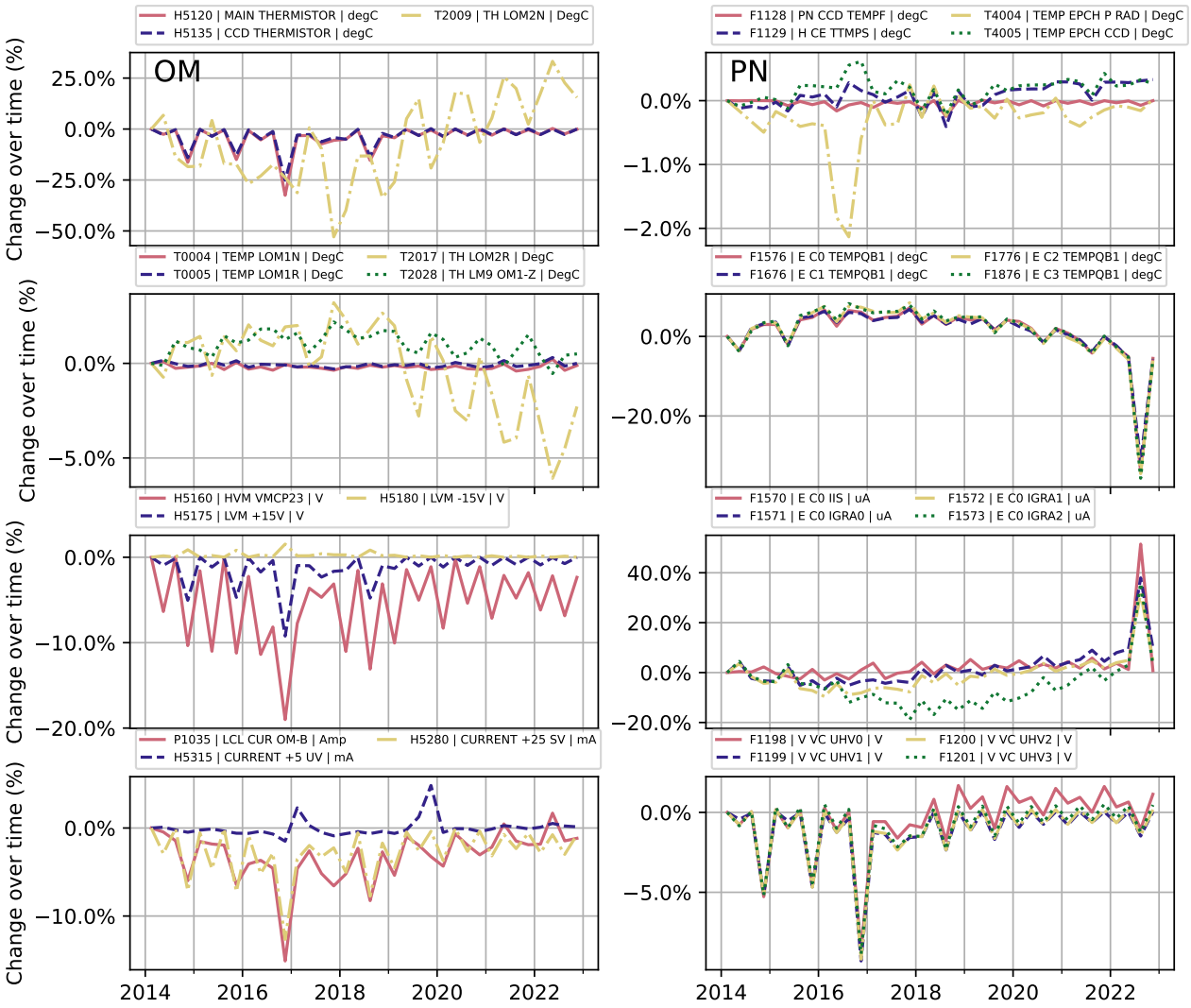
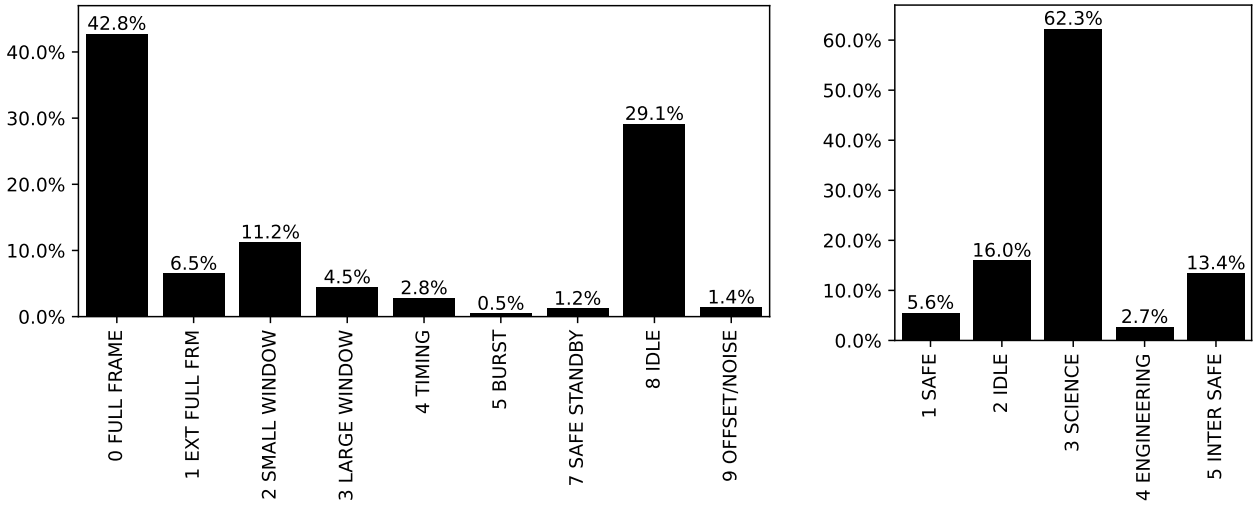


Figure 3.3: Quarterly aggregated mean values for various telemetry channels in OM and PN, showcasing various level of stationarity and some mildly visible yearly seasonalities. It should be noted that the mean value can be skewed by outliers, but better captures the extremes of a channel than the median.

3.1.3. Instrument modes

Both OM and PN have a primary mode channel which gives the active operating mode of an instrument (previously described in Section 2.1.2). The frequency of use for each mode is shown in Figure 3.4. Both instruments operate in science modes (modes 0-5 for PN and mode 3 for OM) for over 60% of the time but are not operational during periapse passes to protect its instruments from the Van Allen radiation belts. This property is not shared with telecommunications or navigation satellites, which may continue to operate their instruments throughout their orbit. The primary operating modes can significantly influence the behaviour of telemetry channels as shown in Figure 3.5. A number of auxiliary modes and statuses are also present, such as the filter wheel setting described in Section 2.1.2. Figure 3.6 shows the connection between a temperature channel of OM and the status of its four heaters. As telecommands are not available in this thesis, modes and status channels

become extra important for this project as they are the only ”human-in-the-loop” input.



(a) PN FD126. **(b)** OM H5395.

Figure 3.4: Mode usage frequencies for the primary mode channels of PN and OM. Aggregation is performed over all data between 2014 and 2023. Modes with less than 0.1% usage are not shown.

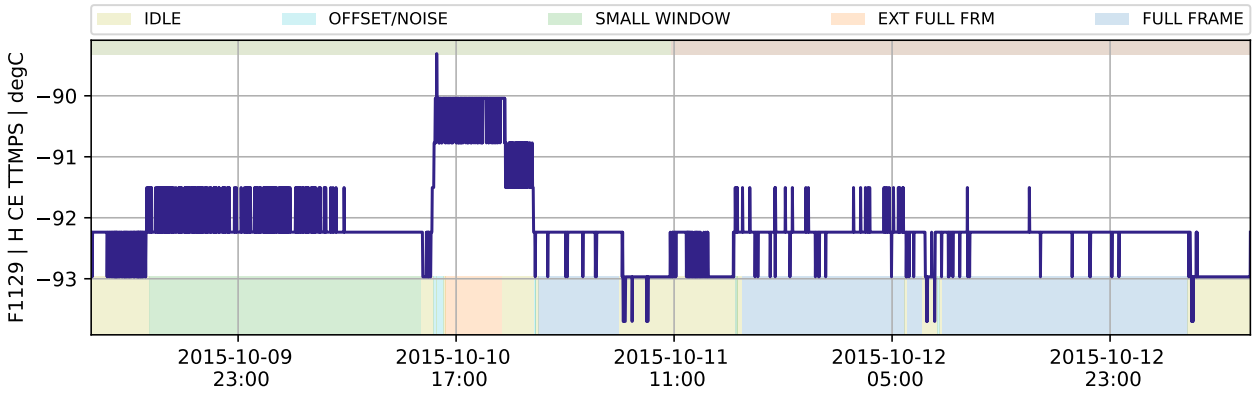


Figure 3.5: A PN temperature channel F1129 over two orbits showing its connection to the primary instrument channel FD126. The channel tends to have a lower temperature during the IDLE mode and a higher temperature in EXTENDED FULL FRAME.

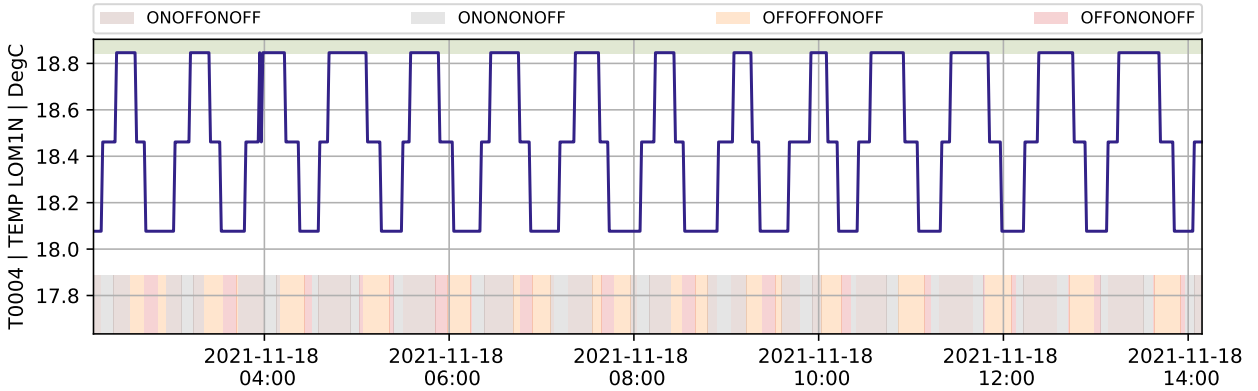


Figure 3.6: An OM related temperature channel T0004 over 12 hours, showing its relationship with the encoded heater status H5240. The H5240 encodes the whether each of the four heaters are ON or OFF.

3.1.4. Eclipses

Figure 3.7 presents all eclipses recorded between January 2014 and July 2024 and illustrates how they are clustered into ”eclipse seasons” of varying lengths (previously described in Section 2.1.3). Due to a cooler thermal environment and a lower power input, eclipses require special operations to keep the spacecraft and its instruments safe. This can be seen directly in Figure 3.8, which shows several channels reaching or approaching their global maxima and minima. A number of subsystems are pre-emptively heated to accommodate uncontrolled cooling during the eclipse. Although the eclipse itself is brief, its effects are observable for hours before and after the event.

The often extreme behaviour of telemetry channels during an eclipse ultimately has a large impact on the anomaly detection methodology and the topic is dealt with extensively in subsequent chapters. In any other context the observed behaviour would be marked anomalous, but because they are known and planned, they should be described as rare nominal events by the definition from ESA-ADB (see Section 2.2.4).

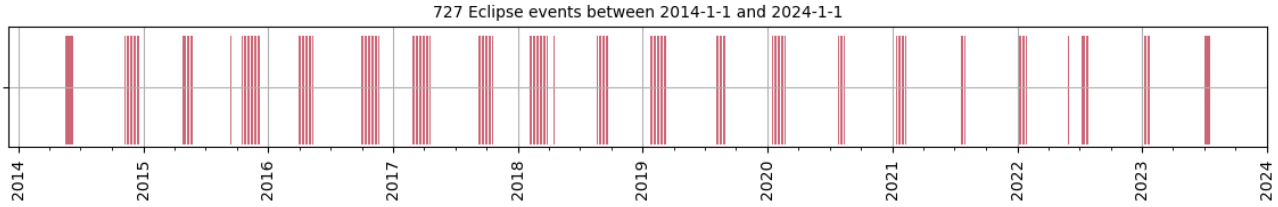


Figure 3.7: Eclipse events between 2014 and July 2024 highlighting the frequency and varying length of eclipse seasons. Bars are exaggerated for clarity.

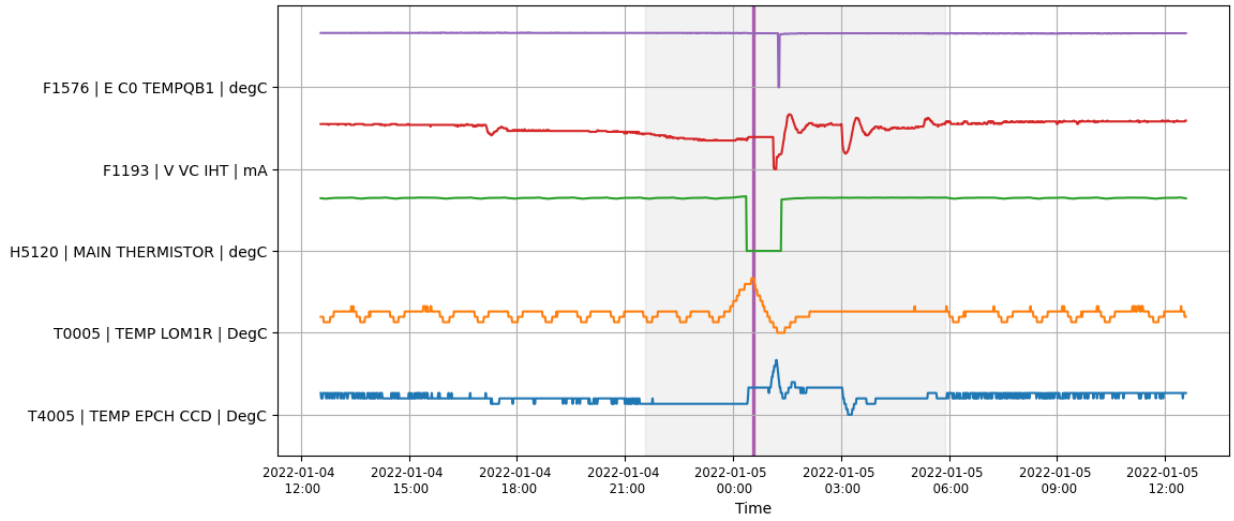


Figure 3.8: An example eclipse event showing the behaviour of various OM and PN channels. The actual recorded eclipse event is highlighted purple. The approximate area of effect is highlighted grey.

3.1.5. Availability of telemetry data

While telemetry data commonly contains gaps due to communication or issues within a satellite, the proportion of missing data is higher in XMM-Newton due to a requirement for continuous communication and a lack of onboard mass storage capacity. Figure 3.9 shows the extent of data gaps and how there is an increased number of gaps between 2016 and 2019, due to a lack of complete ground coverage in this period (see Section 2.1.4).

A closer look in Figure 3.10 shows how instrument specific channels (e.g. H for OM) have more missing data than channels of other subsystems (e.g. Thermal (T) and Power (P)) and this can also be seen directly in Table 3.2. At a later point in the project it was realised that problems affecting the spacecraft as a whole, most prominently those related to the spacecraft command module, occasionally affect the recording or transmission of instrument related channels. Similarly, problems that affect an instrument at a higher level are sometimes also manifested as missing data in more localised telemetry channels, as can be seen in Figure 3.11.

The volume of missing data is far larger than that of ESA-ADB *Mission1* and *Mission2*, which count a total of 4 communication gaps across both missions. The situation in XMM is much closer to that of the discarded *Mission3* which includes a total of 553 communication gaps and invalid segments over an 8 year period. Similar

to eclipses, segments of missing data are treated as rare nominal events and pose a significant challenge for the remainder of the project.

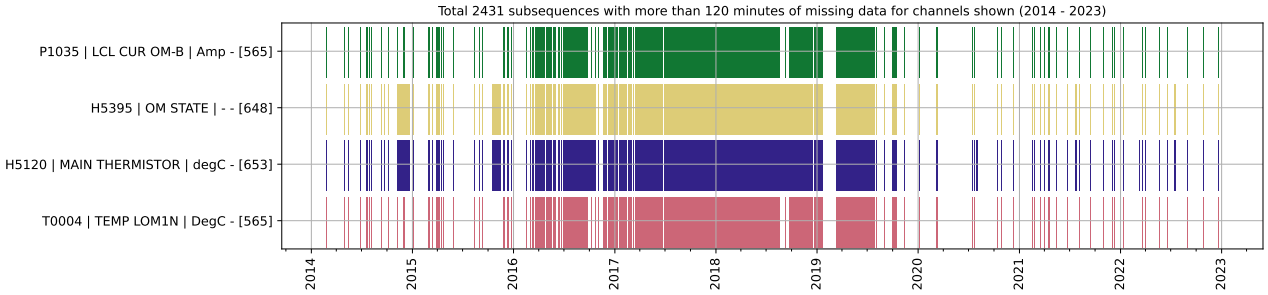


Figure 3.9: All sections with more than two hours of missing data between 2014 and 2023 for selected channels in OM. Total is calculated across channels and the scale of bars show real durations.

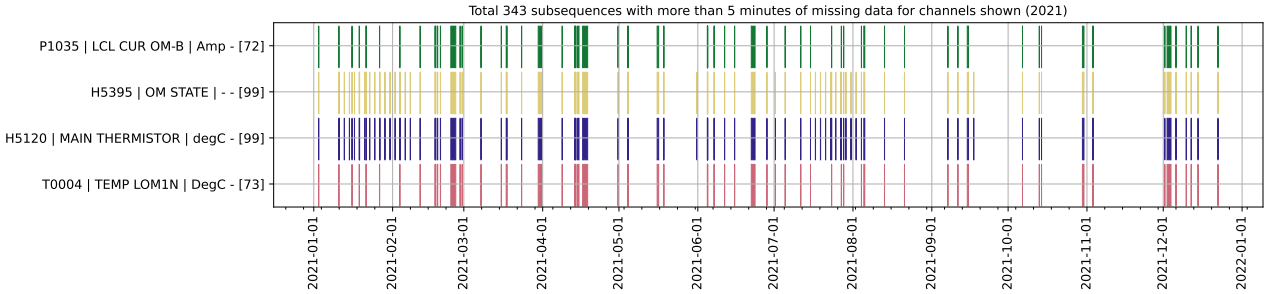


Figure 3.10: All sections with more than 5 minutes of missing data in 2021 for selected channels in OM. The total is calculated across channels and the scale of bars is exaggerated for clarity.

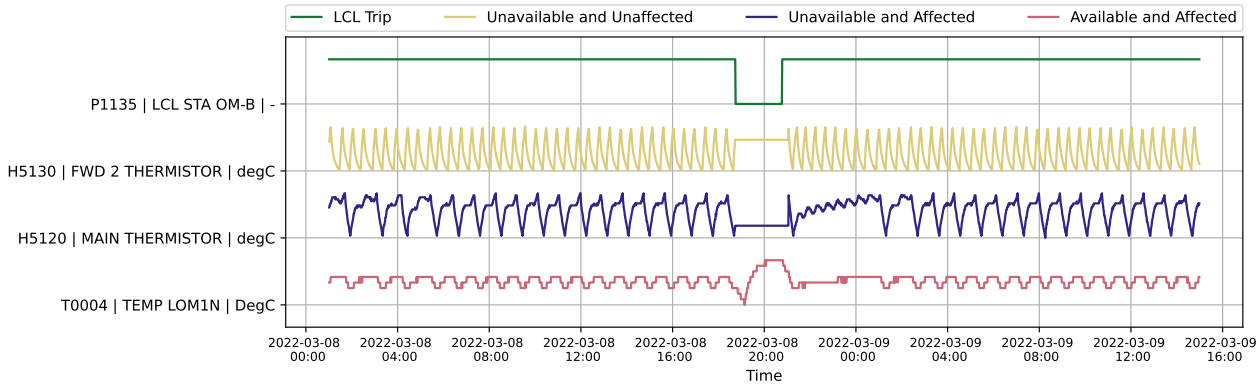


Figure 3.11: The effect of an OM Latching Current Limiter (LCL) trip, showing various effects across OM related channels. Event is retrieved from anomaly report SC-143

3.2. Existing anomalies

Access was provided to anomaly reports related to the spacecraft bus (marked SC) and its instruments (marked IOPS), both shown in Figure 3.12. These reports, discussed further in Section 4.1.3, only contain the most serious anomalies. The IOPS collection, started in 2019, shows that OM has the most recorded anomalies while PN has among the fewest. Each event was checked manually, finding that only few (4 for IOPS, 8 for SC) that directly or indirectly affect channels analysed during the project (primarily temperatures). For example, a large portion of OM related issues were found to be related to processing and memory issues. One particular anomaly: SC-143, a Latching Current Limiter (LCL) trip already shown in Figure 3.11, was found to be particularly useful as a benchmark example and is used throughout this report. The remaining events will be compared to the results in Section 6.2.1.

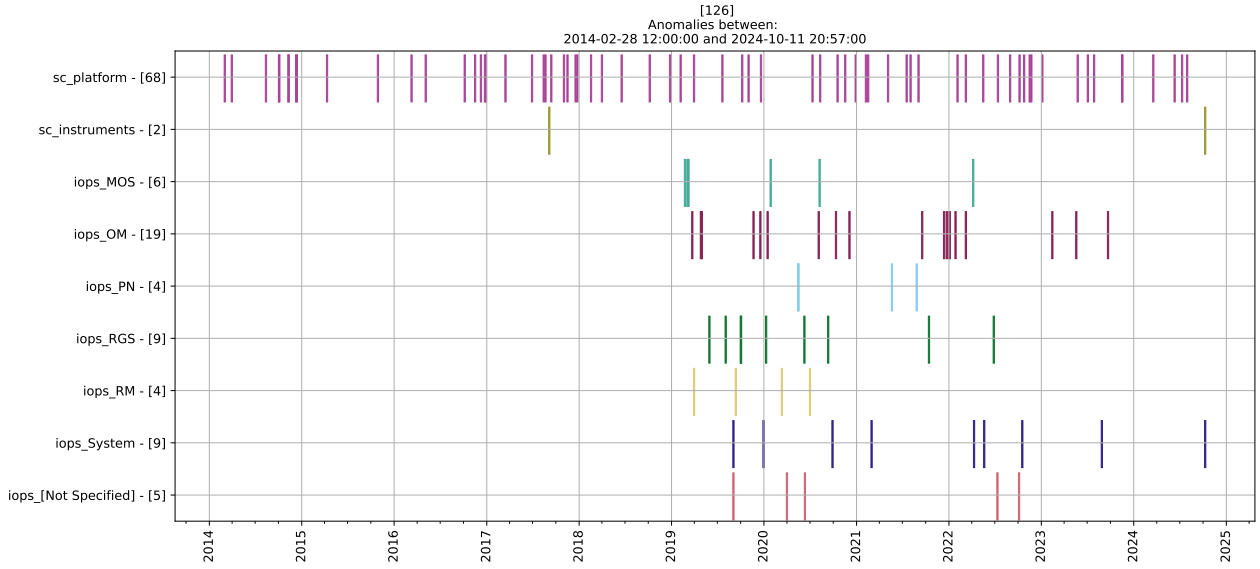


Figure 3.12: Reported instrument (IOPS) and spacecraft (SC) anomalies since 2014. IOPS reports have only been catalogued since 2019. Scale of bars is exaggerated for clarity.

3.3. Consequences for methodology

The findings presented in this chapter have varying consequences for the methodology. To start, the appearance of channels, their availability and stationarity are used to select targets for anomaly detection. Additionally, the discovery of the properties shown in Figure 3.11, lead to analysis on channels of the thermal subsystem (T), ultimately leading to positive results. Instrument modes contain many non-ordinal encodings which must be processed in a special way before being ingested into machine learning models. The volume of missing data and eclipses, which can be considered rare nominal events also require special attention. All these properties are covered in the data preprocessing stage in Section 5.1.

Finally, the high volume of missing data and the lack telecommands in XMM data are all characteristics shared with ESA-ADB *Mission3*. That dataset was ultimately discarded as a benchmark due to having only a small number of trivial anomalies and due to its high volume of communication gaps and invalid segments. Additionally, the regular recurrence of eclipses, which can be considered rare nominal events pose a challenge for TSAD algorithms. This all indicates that XMM telemetry data could be considered a potentially challenging case for anomaly detection and for the generation of an anomaly benchmark. As other satellites might share similar traits, the existence of a challenging anomaly benchmark could foster innovation in ML-based TSAD methods, which may benefit XMM and other satellite operations.

4. Methodology

This chapter provides an overview of the resources, tools, methods and planning used to perform this thesis project. Where relevant, arguments are provided for the chosen method, leaving finer details for subsequent chapters that explain the construction of the proposed anomaly pipeline. Section 4.1 provides the resources provided by the XMM Team, including an overview of data sources, and describes the software tools and computational resources used. Section 4.2 provides a definition of what is considered an anomaly, the approach used to detect them, and an indication of which telemetry channels are considered best for the search. Finally, Section 4.3 gives an overview of how the project was managed and planned.

4.1. Resources and tools

This section presents the resources and tools used to perform the thesis, with an overview of telemetry data in Section 4.1.1, the software tools and computational resources presented in Section 4.1.2 and the support and existing anomalies provided by the XMM Team at ESAC in Section 4.1.3.

4.1.1. Telemetry data

This thesis project is fundamentally a data science and engineering project and is made possible by XMM Newton telemetry data, newly accessible through ARES and ATAS. This telemetry data forms the prime resource for this thesis. Some auxiliary data sources are used to generate additional covariates, for pre- and post-processing and for visual analysis. An overview of the various data sources is given in Table 4.1.

Table 4.1: Overview of data sources used in the thesis. Sources of the public XMM spice kernels and its prerequisites are provided in Section 5.1.2.

Data source	Availability	Information provided	Format
ARES	Private	XMM Telemetry data, retrieved through ATAS.	Pandas DataFrames
ATAS configs	Private	Which channels belong to each Instrument, mode definitions, channel descriptions and units.	.yaml configs, excel spreadsheets
XMM SPICE Kernels	Public	Orbital data.	.spk files
XMM eclipse record	Private	Eclipse umbra and penumbra times.	Fixed-width formatted table files
XMM revolution numbers	Private	Beginning and end times of each revolution.	Fixed-width formatted table files
XMM out-of-limits data	Private	Out of limits events.	Fixed-width formatted table files

Many of the telemetry channels used are available from the early 2000s up to now, potentially allowing for over 20 years of raw data. There are however some soft limits to how much data can be used. As an example, the operational regimes of several instruments had been changed between the launch and 2004. Additionally, the automation efforts beginning in 2008 have significantly improved the operational efficiency of the mission, reducing scientific down time. Ultimately, the most significant limitations arise from the use of auxiliary data sources. As explained in Section 2.1.3 and shown in Section 3.1.4, eclipses have a considerable effect on the satellite's functioning, requiring special operations. Eclipses affecting XMM are not recorded in ARES and have been provided in a separate archive, with eclipse events from January 2014 to July 2024 (thesis start). As their inclusion to project, both as a covariate and a resource to filter out rare events is crucial, January 2014 has been set as a hard lower limit on the data.

Similarly, orbital data in the form of SPICE kernels (SPK) define the upper limit. ESAC publicly offers two SPKs: one derived from measurements, spanning January 2017 to February 2023, and another constructed from *JPL Horizons* ephemeris data, covering December 1999 and February 2023. The latter, which provides a broader range was ultimately selected. The resulting 9-year span of data from the 1st of January 2014 to the 1st of February 2023 was considered to be sufficient for the thesis and is comparable to ESA-ADB (14 and 3.5 years) in size. The inclusion of more recent orbital data can be considered in future work.

As of the start of the thesis, the ARES archive for XMM primarily focussed on instrument telemetry data for

interpretation and analysis by the instrument engineers. Telecommands used to control the spacecraft, along with platform related telemetry such as pointing and radiation monitor data was not available. Their inclusion could be considered in future work.

4.1.2. Software tools and computational resources

As discussed in Section 2.4, the bulk amount of software development is performed on the ESA Datalabs platform: this enables access to ATAS and XMM telemetry data and provides access to computational resources. Datalab's cloud computing architecture means the total computational resources have varied throughout the project. Generally, at least 16 compute cores, over 64 GB of RAM and a shared Nvidia L4 GPU were available for use. In replication, a reader should use a device with a modern CPU, at least 64 GB of RAM. The use of a modern GPU is recommended to reduce training times of deep learning models.

Software development starts at the ATAS Python library. Python is commonly used for data science applications and has been widely adopted by the anomaly detection field with many authors open-sourcing the code for their TSAD algorithms in Python. However, instead of collecting and applying existing TSAD algorithms, the choice was made to use an anomaly detection framework. Because there is currently no 'industry standard' anomaly detection framework, *Darts* [66] was chosen as it is currently used by the ESA Science Directorate. Its use comes with a number of advantages and disadvantages:

- **Advantage** — Darts is compatible with the Pandas library used for data preprocessing and post-processing.
- **Advantage** — Darts has extensive support for multivariate data and covariate time series. This is useful for the inclusion of data like instrument modes and eclipses. Darts also supports per channel anomaly detection.
- **Advantage** — Darts handles the much of the data validation and data manipulation required for forecasting algorithms. Covariate and target channels as well as the different time-lagged transformations for input data are handled automatically.
- **Advantage** — The process to apply forecasting and anomaly detection is syntactically the same for all algorithms.
- **Advantage** — Darts has some built-in compatibility with SHAP, used for explainability analyses.
- **Disadvantage** — Darts is largely a forecasting framework. While an interface with *PyOD* [67] exists that allows for the use of a large number of algorithms mentioned in Section 2.2.2, deep reconstruction methods are largely unavailable.
- **Disadvantage** — Despite being designed for forecasting and TSAD anomaly detection, Darts does not include implementations for a number of popular forecasting algorithms available in literature. Manual implementation of these algorithms is possible at the expense of the user but was avoided due to time constraints. The framework does include many of the common underlying architectures used in published algorithms, such as XGBoost and LSTMs.

The compatibility with multivariate data and covariate channels is a big benefit in the context of this thesis. Additionally, conclusions from Section 2.2.2 hint at an approach that applies simpler algorithms instead of models with "many moving parts" such as complicated graph neural networks and generative models.

4.1.3. Support and existing anomalies provided by the XMM team

An all-encompassing manual of information on the telemetry data was not available, instead instrument engineers were consulted on occasion to provide deeper insight on the meaning of specific channels.

The first part of the thesis (up to the midterm) was carried out without the use of existing anomalies. This left the possibility of a blank-sheet design of the anomaly detection systems and forced an exploratory approach that covered a large amount of data with minimal experience. A subsequent change in strategy enabled the heavily parametrised systems developed in the thesis to be refined and completed through comparison with existing anomalies.

Existing anomalies were provided in the form of reports from the Anomaly Report Tracking System (ARTS). Basic information such as the number of reports and their distributions have already been shown in Section 3.2. Only a small portion of anomalies are documented in the system, as it was only adopted relatively recently for XMM instrument telemetry. It primarily contains the most serious events, while spacecraft operators record occurrences of well-known anomalies and other minor issues elsewhere. These additional sources were not directly available for a number of reasons. Nevertheless, the small number of relevant reports proved sufficient to tune detection systems and also allowed for a comparison with anomalies detected during the thesis. Discussion with instrument engineers near the end of the thesis allowed for further confirmation of detection results.

4.2. Approach

This section provides a global overview of the approach used to solve the research questions presented in Chapter 1. In Section 4.2.1 the chosen anomaly detection method is presented. Section 4.2.2 then presents the definition of an anomaly. Finally, Section 4.2.3 discusses where these anomalies can be found.

4.2.1. Anomaly detection method

The primary research question **RQ1** of this thesis is to find a suitable approach to construct a dataset of anomalies in XMM. A lack of previous experience with XMM's telemetry and spacecraft telemetry as a whole as well as the limited time frame of this thesis makes finding the optimal approach unlikely. Additionally, there are multiple approaches that may achieve a desirable result. Instead, it was chosen to develop an initial approach and subsequently discuss its efficacy afterwards. That discussion is provided in Chapter 7.

The use of the Darts framework lends itself to a semi-supervised forecasting approach. A benefit of many such approaches is the native ability to utilise covariate channels, such as modes, telecommands and statuses, which are common in spacecraft telemetry. ESA-ADB authors note this as a should-have feature for any TSAD algorithm applied to spacecraft telemetry [18]. The anomaly detection features in Darts also best support a direct-detection approach, the most popular method found in literature. The anomaly detection features in Darts are built around providing detections per channel, an important property of the ESA-ADB dataset.

The process to go from raw telemetry data to a preliminary dataset of anomalies using forecasting methods can be divided into four stages, broadly described below:

1. **From raw telemetry and auxiliary data to machine learning-ready data** — As described in Section 2.2.2, preprocessing is required to convert raw telemetry data into a format that is digestible for machine learning algorithms. Additionally, auxiliary data like eclipses and orbital data must be incorporated.
2. **From machine learning-ready data to forecasted telemetry data** — Once ready, the processed data can be fed to the forecasting algorithms. One of the conclusions from the benchmark review in Section 2.2.3 is that no one TSAD algorithm performs best on all anomaly datasets. Therefore multiple models will be trained to select the best algorithm.
3. **From forecasted telemetry data to anomaly detections** — An accurate reproduction from the forecasting model can then be used to find a forecasting error, which can then be modified using one of several existing scoring functions to produce an anomaly score. A threshold can be applied to retrieve anomaly detections.
4. **From anomaly detections to a dataset** — Post-processing may be done to add supplementary information to each detection. The anomalies must then be stored in a format which can be used for an anomaly benchmark.

The method used for forecasting, shown in Figure 4.1, involves target and covariate channels. Target channels are those that are forecasted and subsequently scored for anomalies. Covariate channels are used by the forecasting model for additional context to improve performance. It should be noted that this method is not true forecasting: while the autoregression performed uses predicted target channels to build subsequent predictions, the covariates provided to the model are always the true values. While this is a sensible approach when ingesting human-in-the-loop channels like modes and system-independent channels like orbital data, other subsystem telemetry channels require more thought. Introducing those channels as covariates may allow the forecasting model to perform better over various seasons but they may also aid the model to an extent that it is able to predict all behaviours too well, including anomalies. As such, special care was taken to avoid providing quasi-duplicate channels as covariates. Additionally, forecasting models perform worse when trained with irrelevant data which may introduce additional noise.

In initial plans, the detected anomalies would subsequently be analysed using explainable AI as part of **RQ2** with the aim of understanding the anomaly and its cause. As mentioned in Chapter 1, work related to explainable AI could not be completed. A number of implementations using SHAP have been documented in Section 5.5 to facilitate future work. Similarly, the plans included discussions with engineers for feedback and confirmation of the anomalies. Unfortunately, their limited availability meant this was only done to a limited extent at the end of the thesis, which prevented iterative improvement.

A combined overview of the steps discussed is shown in Figure 4.2. The methodology shown was implemented in the form of an anomaly pipeline, which is detailed extensively in Chapter 5.

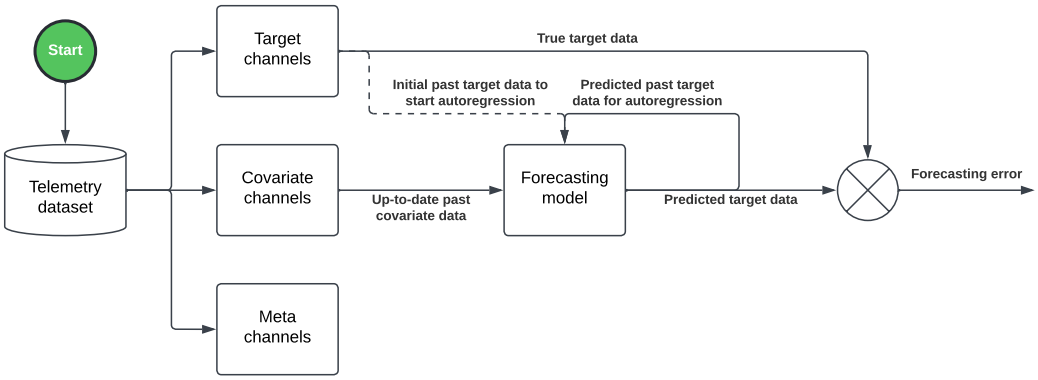


Figure 4.1: The forecasting method applied with the Darts framework. Meta channels include revolution numbers and dates, used to visualise and analyse results.

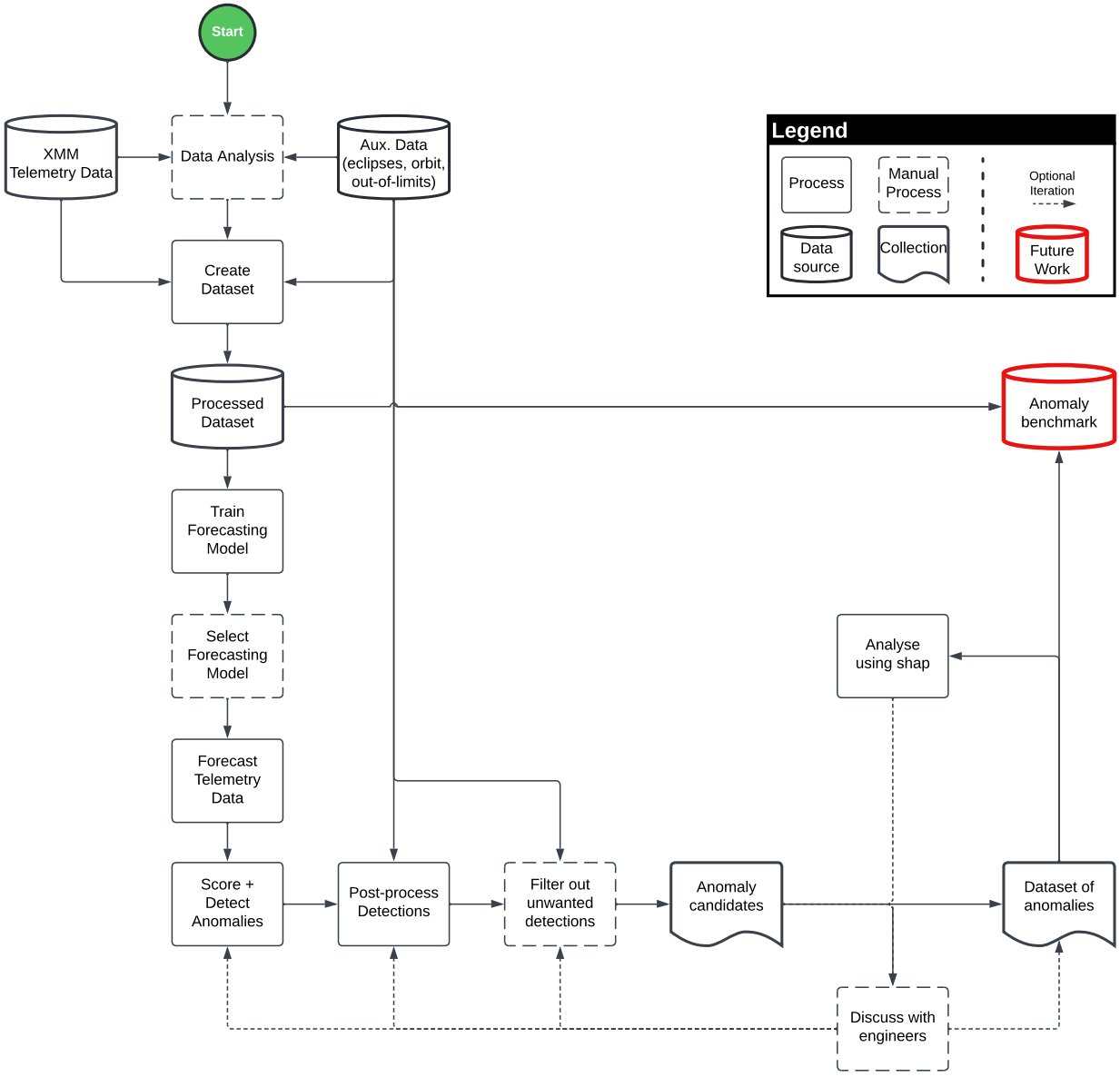


Figure 4.2: Simplified overview showing the steps to go from raw data to anomaly detections, SHAP analysis and an eventual anomaly benchmark.

As discussed in Section 2.2.2, semi-supervised models require training data to be free of anomalies. While some known anomalous segments are removed from the data (see Section 5.1.5), a larger number of initial anomalies could have been collected (e.g. using unsupervised methods) before proceeding to semi-supervised learning. Such a step is recommended for future work in and discussed further in Chapter 7.

4.2.2. Anomaly definition

Similar to ESA-ADB, events detected by TSAD algorithms in XMM may be considered anomalies or rare nominal events. As an example, XMM's behaviour is clearly and significantly affected when experiencing an eclipse. The occurrence of an eclipse is known and planned for however, making the label rare nominal event more appropriate. Because these events are known and recorded, detections that overlap with an eclipse can be annotated automatically. Similarly, out of limits events are recorded and communication losses can be inferred from the telemetry data. Marking other detections as rare nominal events is more difficult however, as they may not be recorded or may not be directly discernable from accompanying commands or covariates. Instruments may for example be tested or calibrated occasionally.

Additionally, it should be noted that not all anomalies are equally interesting to an instrument engineer or ground operator. Events like brief spikes are commonly either glitches in data acquisition or are swiftly dealt with by the automatic FDIR system. Over time, procedures are implemented to automatically deal with common and well-known anomalies. Recording every inconsequential anomaly is time consuming and anomaly reports are often only created for serious or novel anomalies.

Ultimately, only an instrument engineer can derive whether an event is truly a rare nominal event or an anomaly. Because discussion with the instrument engineers was limited during the thesis, the choice was made to mark all desirable detections as anomaly candidates. Not all of those are useful however, as a model may falsely mark a section as anomalous or fail to detect some anomalies entirely.Table 4.2 shows a matrix of possible detections states and their desireability. A TSAD algorithm is not perfect and in practice, a sizeable portion of detections are false positives. Some of these false positives may be easy to spot, such as when they are caused by bad forecasting. Others require the experienced eye of an instrument engineer to be confirmed.

Table 4.2: Matrix of possible anomaly detections.

	Anomalous	**Not Anomalous**
Detected	**True positive** — Anomaly or rare nominal event — To be confirmed by engineer — Add to dataset	**False Positive** — Bad prediction or badly scored — To be confirmed by engineer — Remove from dataset
Not Detected	**False Negative** — Anomaly is not found — Forecaster predicts anomaly / scorer misses anomaly — May be added in future versions of catalogue	**True Negative** — Normal behaviour — 99% of data

False negatives also require special attention. Consider the following hypothetical scenarios that yield false negatives:

- *Training data for the forecasting model includes a known common anomaly type that is not filtered out. The forecaster might then be able to predict the anomaly, leading to a missed detection.* In such cases, iteration in the training method is required to successfully detect these anomalies.
- *Multiple anomalies of the same type exist. Some are detected but others are not.* In such cases, the undiscovered events may then be found using pattern-matching techniques and an improvement in the anomaly detection method would be required.

Each case requires iteration and/or expert knowledge to confirm, which is not possible due to the time-limited scope of the thesis. Additionally, a larger collection of anomalies would be required to accurately check false negative performance. As such, these events can only be treated to a limited degree within this thesis.

4.2.3. Where to look for anomalies

Channels accessed by ATAS to generate reports number in the hundreds. While the XMM team did not directly recommend specific channels to look at, they did suggest to look at telemetry for PN because it is the most

important, and OM because it is most error prone. The latter feature was also found to be true in Chapter 3.

Beyond recommendations, important factors to look at are the availability of data, how fast their values change over time and resolution of the measurements (e.g. a temperature sensor that only has two recorded values). As an example many channels that measure current in PN have a very noisy and fast moving signal. In those channels it becomes more difficult to detect an anomaly. Ultimately it was chosen to first look a temperatures, since these channels usually change very slowly over time, often have a common periodicity (especially in the case of thermistors) and are critical enough to have widespread availability. Unlike instrument-specific channels, those related to the thermal subsystem are often still available even during a malfunction in the instrument, providing full coverage.

The XMM team explicitly stated a disinterest in short, spike type point anomalies as these usually indicate a glitch and can be caught easily by existing out-of-limits systems.

4.3. Planning

Near the start of the thesis, a long-term plan was devised to keep the project on track. A rigid waterfall-style approach was avoided due to the exploratory nature of the subject as well as iterative nature of software development. For example, new discoveries made while exploring the data and feedback from instrument engineers were expected to drive iterations of the software tools and methodology. Instead, the project was divided into five successive research phases, each with a number of Work Packages (WP):

Phase 1: Literature review, data exploration, and research definition

- **WP1 -** *Literature review* — Exploration of relevant topics in literature. Chapter 2 presents a condensed version of the findings.
- **WP2 -** *Develop data tools* — Early development of a data tool to facilitate data exploration.
- **WP3 -** *Early data exploration* — Completed to inform the design of the initial methodology and the pipeline. The results form the basis of Chapter 3.
- **WP4 -** *Research definition* — Includes the definition of initial research questions and project planning
- **WP5 -** *Development of initial methodology* — Includes the design of the anomaly detection approach presented in this chapter, as well as early testing of the Darts library.

Phase 2: Anomaly detection

- **WP6 -** *Anomaly detection implementation* — Develop and test the tools required to perform anomaly detection as described by the methodology in this section.
- **WP7 -** *Formalise anomaly pipeline* — Combine the individual tools into a pipeline.
- **WP8 -** *Perform anomaly detection* — Use the pipeline to detect anomalies in XMM telemetry data.
- **WP9 -** *Analyse results* — Analyse the total collection of detections, discussing them with instrument engineers, and comparing them to existing anomalies.

Phase 3: Explainable AI and writing

- **WP9 -** *Implement Explainable AI methodology* — Develop the tools required to perform analysis with SHAP.
- **WP10 -** *Perform analysis with explainable AI* — Analyse detected anomalies with SHAP.

Phase 4: Completion of results and thesis report

Phase 5: Finalisation of the thesis (All steps after the green-light review)

An informal scrum-inspired method was then used for day-to-day planning. New actions discovered during research, software development, and from supervisor or instrument engineer feedback were added to a backlog. Each item was then assigned a maximum completion date of either one or two weeks into the future, in line with the weekly progress meetings. This method proved effective and was used up to the final thesis writing stage.

The planned duration for each phase was chosen based on the expected difficulty of the work packages as well as personal strengths (e.g. experience with data science and software development) and weaknesses (e.g. slow writing, inexperience with spacecraft telemetry and operations). Individual work packages did not receive precise duration estimates as many were expected to be performed in parallel. Figure 4.3 compares the planned and true durations of each research phase. Although phase 1 and early stages of phase 2 progressed quicker than expected, inexperience with telemetry data as well as unforeseen constraints on expert support delayed the initial discovery of anomalies. The introduction of ARTS reports enabled the first real detections

to be made which subsequently allowed for the completion of the pipeline. The unexpected delays with phase two also resulted in much of the work related to phase 3 (and thus **RQ2**) being cut.

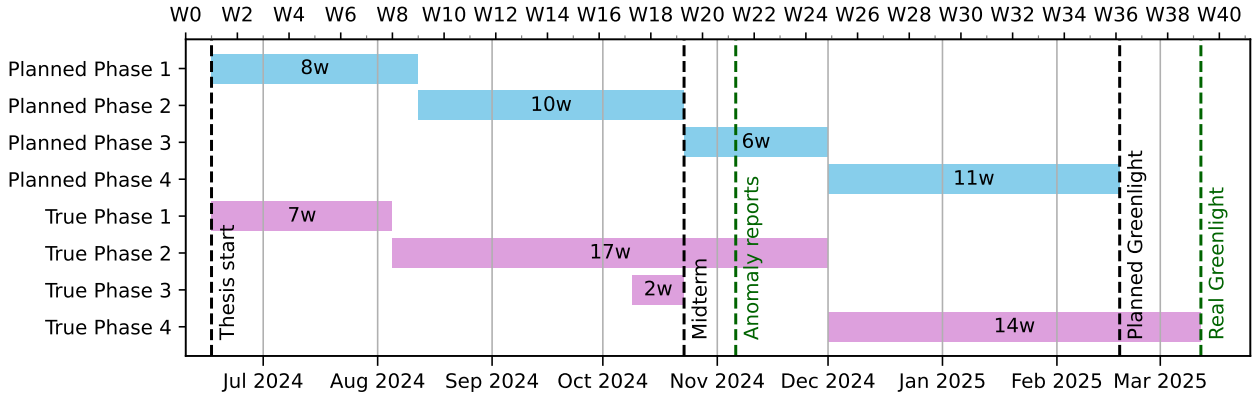


Figure 4.3: Thesis timeline with the planned and true durations of each research phase. Days off and holidays are not included.

5. From data to anomaly: The Anomaly Pipeline

Section 4.2 outlined a broad list of steps to go from raw telemetry data to anomaly detections. This chapter presents their implementation as integrated into an automatic anomaly pipeline, shown in Figure 5.1. Section 5.1 presents the various preprocessing measures used to convert raw telemetry data into a final format that is ingestible by machine learning models. Next, Section 5.2 shows how this data is ingested into forecasting models yielding a forecasting error used for anomaly detection. Comparisons are then made to analyse the performance of various forecasting models and how the configuration of a dataset can affect model output. In Section 5.3, a number of scoring functions are applied to the raw forecasting error to analyse the potential benefits. A threshold is then applied to the chosen scorer to retrieve anomaly detections, which are then post-processed in Section 5.4. Although an interpretability analysis with SHAP was originally intended to form part of the pipeline, the development of this segment could not be completed due to time constraints, as mentioned in the introduction. The incomplete SHAP implementation is documented in Section 5.5.

The pipeline is able to perform forecasting model benchmarks. Its many settings are controlled through configuration files, which are split into four abstractions:

1. **Dataset configuration** — controls how the dataset is processed. Most settings are discussed in Section 5.1.
2. **Model configuration** — controls how the forecasting model is created and trained, including its hyperparameters. Multiple models can be passed to be trained in a single run.
3. **Scorer configuration** — holds parameters for the scoring functions and the quantile thresholds.
4. **Pipeline schedule** — collects the above configurations and provides them to the pipeline.

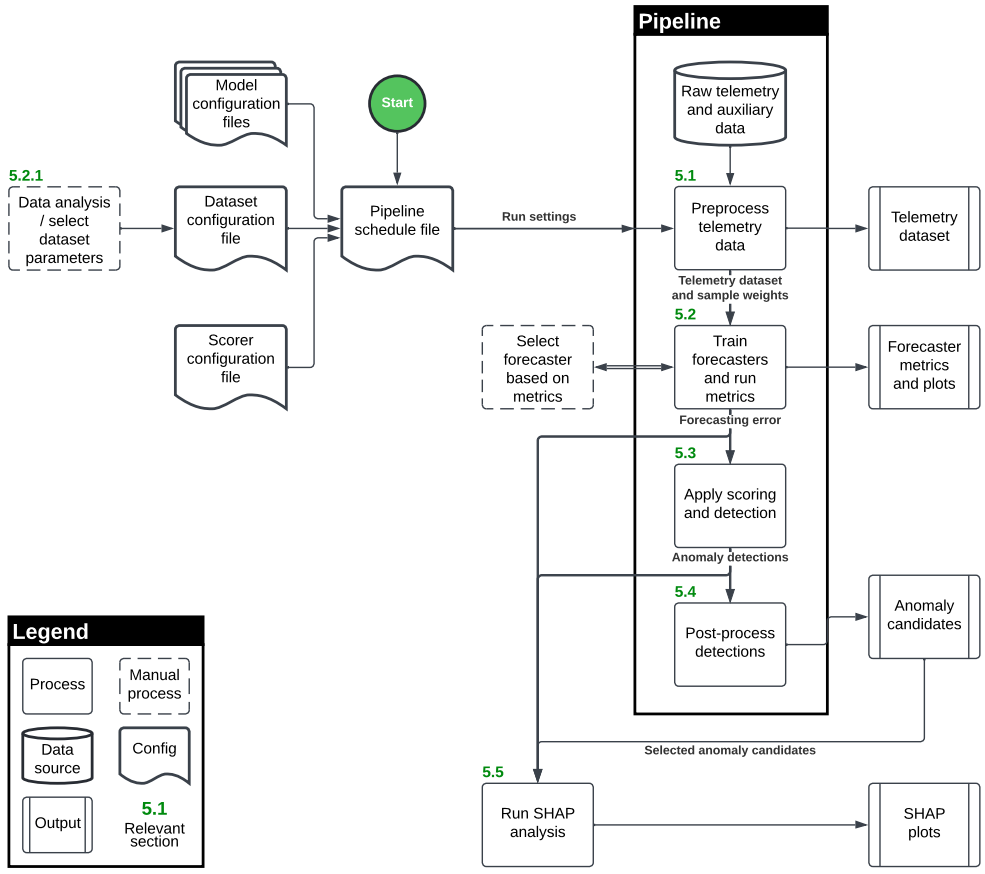


Figure 5.1: Simplified overview of the anomaly pipeline.

5.1. Telemetry data preprocessing

As found in Section 2.2.4, raw telemetry data is innately incompatible with machine learning models. To retrieve and process the data into a format that is digestible by machine learning, a telemetry 'dataset tool' has been developed. The following informal requirements were identified at various points in the thesis:

D1. **Dataset splitting** — To avoid over-fitting, semi-supervised models should be provided with separate training and testing datasets and ideally also a validation set, as described in Section 2.2.2. The tool should be able to split the data into a training, validation and test dataset.

D2. **Collation** — Telemetry data has varying sample sizes and irregular acquisition times which must be converted to regular time intervals by the tool. Discussed in Section 5.1.1.

D3. **Adding auxiliary data** — Some auxiliary data sources such as eclipse dates and orbital data must be combined with the telemetry data by the tool to improve forecasting and allow for a complete picture in analysis. Discussed in Section 5.1.2.

D4. **Scaling** — Telemetry channels have varying units and magnitudes and must be scaled by the tool to ensure compatibility with deep learning based models. Discussed in Section 5.1.3.

D5. **Handling modes** — Some machine learning models are incompatible with categorical data such as modes, which must be converted to a friendly format by the tool. Discussed in Section 5.1.4.

D6. **Removing anomalies from training data** — Semi-supervised models, such as the forecasters used in this thesis, nominally require training data to be free of anomalies. Thus, the tool should eliminate anomalies from the training data. Discussed in Section 5.1.5.

D7. **Future-proofing** — The results of the thesis may be carried over by the XMM team or eventually be used to create a new anomaly benchmark. The tool should be designed for use beyond the thesis. Discussed in Section 5.1.6, the minor remaining limitations are provided here as well.

Point **D1** is trivial, and ultimately solved by implementing functionality that allows the tool to retrieve data for a train, validation and a test set using a provided start and end date for each set. Choosing the actual size for each set is described in Section 5.2.1. The remaining points require additional explanation, provided in the subsequent sections.

5.1.1. Collating telemetry data to regular intervals

As shown in Table 3.2, channels have different sample rates and have portions of missing data. Additionally, data points have imperfect acquisition times. A 'collation' step to cast the data to a regular time steps follows these steps:

1. The user selects a start time and a collation period, e.g. 60 seconds. Sample times are set from the start time, with each subsequent 60 seconds marked as a sample time.
2. For each sample time, a sample period is set between the previous sample time and the current sample time. All points in this sample period are aggregated with the value of the last point being set as the value for the current sample time.
3. Sample times where the sample period is empty, receive the value of the last sample time with a value.
4. Sample times that precede the first data point receive the value of that first available data point.

The process is similar to the "zero-order hold resampling" method used for ESA-ADB [18] and is best understood by looking at Figure 5.2. The difference with ESA-ADB is the use of a varying collation period and start time, ESA-ADB utilises the ideal acquisition times as a collation period and sets the start at the first ideal collation period. A variable collation period is useful to optionally reduce the amount of data to be forecasted, thus reducing TSAD algorithm computation times, but the same result as ESA-ADB can be achieved by carefully selecting the collation period and start times.

An unfortunate consequence of this method is that large periods of missing data will receive the value of the last raw data point. Situations where data is missing may occasionally be caused by errors, where the final value passed on to subsequent missing data points is unrepresentative of normal behaviour. Large sections of missing data, such as communication gaps are marked as rare anomalous events in ESA-ADB. As such, sections which are found to have missing data as are collected into a 'missing data' mask in the collation step. This mask is then used in Section 5.1.5 to limit the influence of missing data in forecasting, in Section 5.3 to reduce the number of missing data points detected, and in Section 5.4 to annotate detections that overlap with missing data.

Instead of aggregating by the last value in a sample period, the data may also be aggregated using a different method, such as the mean, or median. The dataset tool has been implemented to allow for choosing the ag-

gregation method per channel processed. The use of the mean as an aggregation method was briefly explored and has potential to deal with channels that have a low measurement resolution (highly discretised) but was ultimately not used for the final results. The use of a mean aggregation method comes at the cost of unrealistic, 'in-between' values being present in the data.

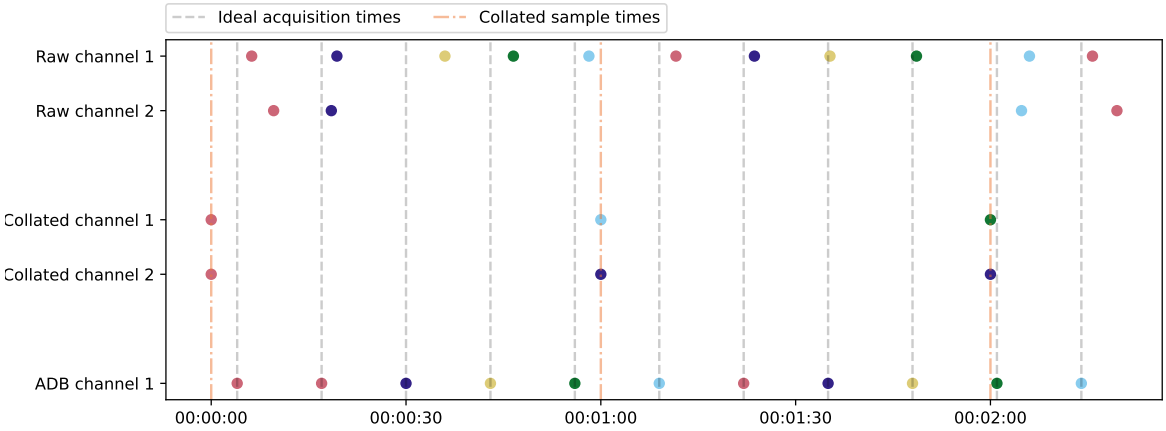


Figure 5.2: Diagram of the collation process performed on two channels with an ideal acquisition time every 13 seconds, starting at 4 seconds. The start time is set at 00:00:00 with a collation period of 60 seconds. The similar method used by ESA-ADB is shown in comparison.

5.1.2. Adding auxiliary data

Eclipse data, revolution numbers and orbital data are not stored and ARES and had to be added separately. As shown in Figure 3.8, eclipses have a significant anomaly-like effect on telemetry data. Some telemetry channels are strongly correlated with the position of the spacecraft relative to the Sun and Earth, as shown in Figure 3.2. Revolution numbers are used for visualisation and the process to add them to the data is trivial.

Eclipse data is presented in the form of start and end times of umbra and penumbra. The satellite is in penumbra when the Sun is partially blocked by the Moon or Earth, in umbra the Sun is completely blocked. These events are added to the telemetry data as an ordinal 'eclipse level' channel. As seen in Figure 3.8, eclipse-related operations precede and exceed far beyond the actual umbra and penumbra events. Ideally, these operations should also be marked in the eclipse channel, but the exact time varies per channel and such data is contained in telecommands which are unavailable for this project. As a compromise, a static buffer is added with points 3 hours before and 15 hours after each eclipse also marked. In the eclipse level channel, points with no eclipse are marked 0, the buffer zones are marked 1 and penumbra and umbra are marked 2 and 3 respectively.

Orbital data is added in 3 forms, distance to the Sun, distance to Earth and the beta angle. The distance to the Sun and Earth are interpolated using the following SPICE kernels:

- **xmm_horizons_19991210_20230223_v01** — XMM Newton kernel based on JPL Horizons — Obtained from ESA ESAC [68].
- **de432s** — JPL Horizons ephemerides kernel used for the XMM kernel — Obtained from NAIF [69].
- **naif0012** — Leap second kernel — Obtained from NAIF [70].
- **naif0012** — Earth kernel — Obtained from IMCCE [71].

The interpolated positions of XMM relative to the Sun and Earth are retrieved every minute in the ECLIPJ2000 reference frame and are converted to magnitude distances. The beta angle β can be retrieved using Equation (5.1), where $\vec{r}_{\odot/\oplus}$ is the position of the Sun w.r.t. Earth, $\vec{r}_{s/\oplus}$ the position of the spacecraft w.r.t. Earth and $\vec{v}_{s/\oplus}$ the relative velocity of the spacecraft w.r.t. Earth. [72]. These magnitudes are then smoothened using a 1st order Savgol filter with a window size of 30 minutes to achieve smooth values for lower collation periods and to remove potential interpolation errors.

$$\beta = \arccos(\vec{r}_{\odot/\oplus} \cdot (\vec{r}_{s/\oplus} \times \vec{v}_{s/\oplus})) \tag{5.1}$$

Higher-quality and more complete orbital data may be achieved by using measurement data directly and should be added in the future with support from the XMM flight dynamics team. However, the data retrieved through Horizons ephemerides is more than sufficient to provide seasonality information to the forecasters.

5.1.3. Scaling data

As discussed in Section 2.2.2 scaling the data is required for a number of machine learning models. Several popular scaling methods, implemented in the Sci-Kit Learn Python library [73] were tested in default settings:

1. **MinMaxScaler** — Proportionally scales and translates each channel to values between 0 and 1. Telemetry in ESA-ADB is provided using this scaler. The scaled channel X_{minmax} can be retrieved using the following formula, where X is a time series of a single channel:

$$X_{minmax} = (X - \min(X))/(\max(X) - \min(X)) \tag{5.2}$$

2. **StandardScaler** — Standardises each channel by translating using the mean and scaling using the standard deviation with the formula:

$$X_{standard} = (X - \text{mean}(X))/\text{stdev}(X) \tag{5.3}$$

3. **RobustScaler** — StandardScaler is sensitive to outliers, such as the point anomalies commonly found in telemetry data. RobustScaler attempts to avoid outliers by instead employing the median to translate and Inter-Quantile-Range (IQR) to scale the data. The scaler uses the following formula:

$$X_{robust} = (X - \text{median}(X))/\text{IQR}(X) \tag{5.4}$$

Figure 5.3 shows how data is transformed by the three scalers. Applying a MinMaxScaler after applying a Robust or StandardScaler has no effect as all three methods apply a linear transformation. Attempts were also made to reduce magnitude of outliers by clipping their values within 2 or more standard deviations, however this approach was ultimately abandoned as changes in operations sometimes meant the nominal range of channel moved outside two or more standard deviations. The effects of the three scalers on the training of forecasting models is compared in Section 5.2.3.

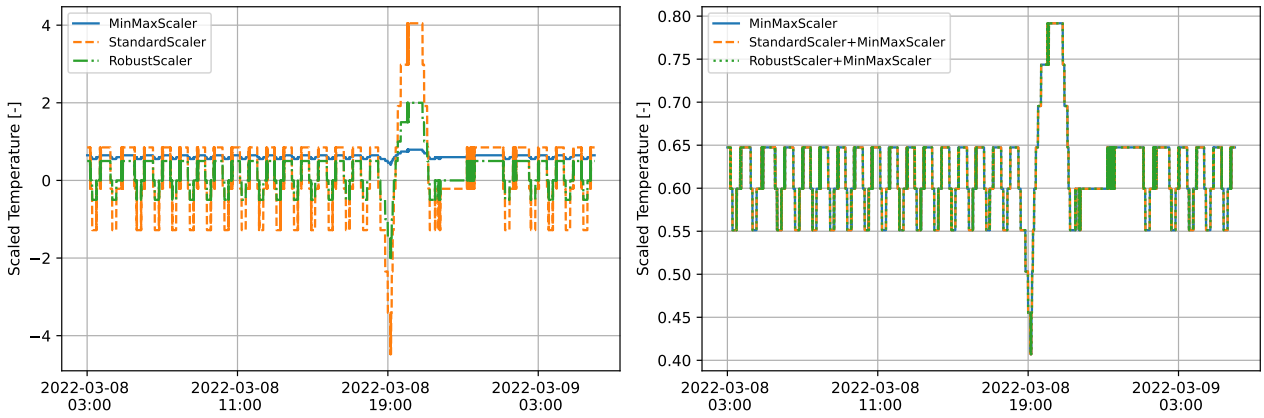


Figure 5.3: The effect of the three scalers on a section of OM telemetry data. The right plot shows the linear nature of each of the transformations, by applying a MinMaxScaler to the remaining scalers.

5.1.4. Handling modes

Modes and statuses such as the primary instrument modes described in Table 2.1 and Table 2.2 are stored as non-ordinal numbers in ARES. Many machine learning models, with the exception of tree-based models like XGBoost [48], can not recognise modes from such channels. A popular method to preprocess categorical data for machine learning models involves using one-hot encoding, where a single channel with multiple modes gets converted to multiple channels each denoting if the a category is active or not. A consequence of this method is that many additional channels are created, which increases the computational cost and may increase overfitting when many categories are present. To process instrument modes in XMM data, multiple modes have been combined to reduce the number of one-hot channels. Table 5.1 list the mode reductions used. As an example, the multiple imaging modes of PN (full frame, small window, etc.) and the temporal imaging modes (timing and burst) have been merged. Rare or unused modes have also merged in singular channels. Not all mode reduction methods are perfect and deeper analysis should be performed to find better combinations based on the resulting physics of each mode.

Table 5.1: Reduction of various modes and statuses to one-hot channels. Groupings are determined by the behaviour of analysed channels. The OM heater status is presented with encodings like 8 = ONOFFOFFOFF, where the on/off status of each of the four heaters is given in a single encoding. Multiple mode reductions for H5240 were used throughout the thesis, based on Figure 3.6, the alternative version might be best, although the original provided better results.

Mode/Status	Reduction	Original #	Reduced #
FD126 (PN primary mode)	imaging (0, 1, 2, 3) timing (4, 5) safe (7) idle (8) other (6, 9, 10, 11, 12)	13	4
FD130 (Filter wheel status)	calibration (1, 3, 5, 7, 9, 11, 15) moving (0) science (2, 4, 6, 8, 10, 12)	16	3
H5120 (OM primary mode)	science (3) idle (2, 4) safe (0, 1, 5)	6	3
H5240 (OM Heater status)	nheaters_on_0 (0) nheaters_on_1 (1, 2, 4, 8) nheaters_on_2 (5, 6, 9, 10, 12) nheaters_on_3 (13, 14) other (3, 7, 11, 15)	16	5
H5240 alternative	all heaters_off (0) heater_1_on (8, 9, 10, 12, 13, 14) heater_2_on (4, 5, 6, 12, 13, 14) heater_3_on (2, 6, 10, 14) heater_4_on (1, 5, 9, 13) other (3, 7, 11, 15)	16	6
HD013 (OM safe/unsafe)	safe (0) unsafe (1)	2	2

5.1.5. Removing anomalous data from training data

As mentioned in Section 2.2.2, semi-supervised models nominally require training data to be anomaly-free. This leads to a paradoxical situation: the algorithm to find anomalies requires anomaly-free training data but they cannot be removed because the algorithm has not found the anomalies yet. The creators of ESA-ADB first ran a number of unsupervised algorithms on the telemetry data to retrieve an initial set of anomalies. These anomalous sections were then removed from a training set provided to a semi-supervised model which found more anomalies which were discussed with satellite operators and removed from a new iteration of training data. This processes was repeated for multiple iterations until no more anomalies needed to be removed [59].

Not collecting a larger initial set of anomalies has already been recommended as an area for improvement in Section 4.2, and is discussed further in Section 7.1. As the ARTS anomalies were only retrieved late into the thesis, they were not removed from the data. This means the only sources for anomalous sequences are: eclipse data, out-of-limits events and missing data. Two methods were considered for removing them from the training data:

1. **Interpolation** — The first method attempted was the removal of sections that contain anomalous sequences and interpolating the resulting gaps. Unfortunately, when considering only simple interpolation methods, the complex repeating patterns of many XMM channels become difficult to interpolate. Additionally, the use of interpolation potentially introduces unrealistic values to the dataset. Finally, training

on interpolated data potentially forces the model to associate synthetic responses to real anomalous behaviour in covariate data. For these reasons a the interpolation approach was abandoned.

2. **Training with Sample Weights** — When training a semi-supervised model, the sample weight is multiplied with the loss function to increase or decrease the effect of specific subsequences in the training set. As such, setting the sample weight to zero in areas with anomalies will effectively make the model blind to anomalous events. This solution is elegant as it allows the whole training data to be passed without cuts.

For now, sample weights are set to 0 in areas with an eclipse, out-of-limit events or missing data and to 1 elsewhere. Future work should incorporate found anomalies into the sample weights, iteratively cleaning up the training data. The use of sample weights also has clear benefits if the anomaly pipeline is applied in a streaming anomaly detection context where the pipeline detects anomalies as new telemetry data comes in. Sample weights can then be used to perform incremental learning, with newer telemetry being provided a higher sample weight.

5.1.6. Future proofing and limitations

Because of the desire to eventually use the results of this thesis to release XMM telemetry data as a benchmark, extra care was made to create a high quality data processing tool. The dataset tool has the following main features:

- **Fast loading and processing times** — The dataset tool makes extensive use of vectorised math and multiple layers of caching to ensure low processing and loading times. Once the raw data is retrieved from ARES, the dataset only takes a few minutes to process years of data and, once processed, the data subsequently can be retrieved almost instantly. For comparison, ESA-ADB notes a time to collate the data of 1.5 hours. While this is acceptable when the data is only processed once (as is the case for ESA-ADB), a user may want to experiment with the data their models run on.
- **High amount of customisation** — A high number of variables, such as the start and end dates, the covariate and target channels as well as the the scaling method and collation settings can selected when processing anomalies into a dataset. All these settings can be set by creating configuration files which set each of these variables in a traceable format.
- **Independent of any existing anomaly detection frameworks** — Section 4.1.2 has listed a number of disadvantages of the Darts framework. To ensure the use of Darts does not limit future work, the dataset tool is made completely independent of Darts, only using the most common data science libraries to produce its data. The dataset tool can effectively be used stand-alone, allowing future users to directly and easily use other anomaly detection frameworks.

The use of ATAS also means that the tool can potentially be used for telemetry of other spacecraft hosted in ARES with minimal modification. A final overview of the dataset tool is provided in Figure 5.4.

Limitations and future improvements

Four functional limitations have been identified which could be resolved to improve the dataset tool in the future:

- **Improved sample weights** — Sample weights should also zero out known anomalies. In addition to those of the anomaly reports, additional anomalies found through the use of the pipeline and others which are known but not included in anomaly reports should also be added.
- **Orbital data** — The dataset tool should be modified to allow for the inclusion of more recent orbital data.
- **Scaling** — The scaling is currently performed locally on the date ranges provided, meaning that data produced for just 2022 might have different scale than a dataset between 2014 and 2022. ESA-ADB handle this by storing the relevant global information (e.g. the min, max, median, etc.) and using those to provide a consistent scaling throughout the dataset. A similar method could be implemented in the future.
- **Hardcoded XMM elements** — While the system is built on ATAS and ARES, potentially allowing expansion to other satellites, some current features, such as the retrieval of auxiliary data and channel information are specifically designed around XMM and require modification to be used for other satellites.

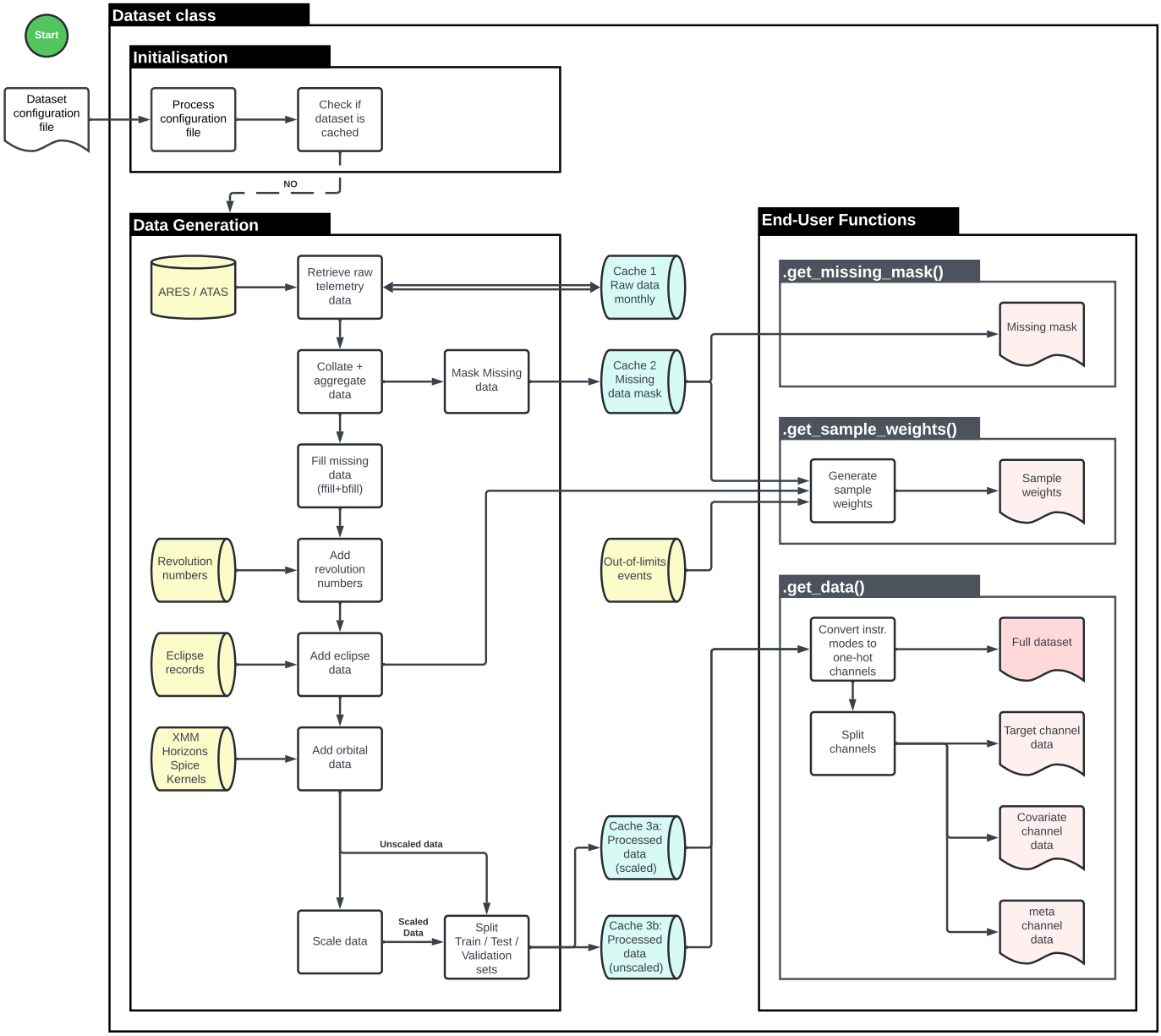


Figure 5.4: Simplified overview of the dataset tool, showing how various data sources are used, how intermediate data is cached and how the final data is retrieved by the user.

5.2. Telemetry forecasting and dataset configuration

This section discusses the steps taken to produce a forecasting error using data generated by the dataset tool. Section 5.2.1 discusses the selection of dataset parameters. Section 5.2.2 goes into detail on the forecasting setup previously presented in Section 4.2. Finally, Section 5.2.3 compares the performance of various models and the effects of various dataset parameters on the forecasting result.

5.2.1. Selecting dataset parameters

The dataset system showcased in Section 5.1 features many processing steps which may be adjusted on a case-by-case basis to improve forecasting performance. This section discusses various considerations made when designing datasets to be run.

Start/end times and training/test cut

Supervised and Semi-supervised machine learning models require a training, test and optionally a validation set, which are often cut into 70%/20%/10% splits (or similar) by practitioners. Training sets often receive the largest cut as machine learning models often require a significant amount of training data to perform well. In the case of this project, the sheer volume of data available and the fact that only the test set may be used for subsequent anomaly detection led to a different approach.

For the large final datasets used to create the results in Chapter 6, the full 9-year range (2014-1-1 to 2023-1-1) is split into two years for training (2014 + 2015), the next half year for validation and the remaining six-and-a-half years as test data. The idea is that the model receives a large set of data, doubly covering the seasonality caused by the orbit of the Earth while still providing a large range of data to perform anomaly detection on. Older data was preferred for training, as instrument anomaly reports only cover events after 2019. For smaller datasets, such as those used for benchmarks in the next section, more recent dates were preferred (2020 onwards) and with at least one year of training data. As a reference, ESA-ADB *Mission1* uses an 81-month/3-month/84-month split.

The approach used in this project does not allow for the training data to be used for anomaly detection. Other publications solve this problem by performing two forecasting runs, e.g. Run A with a first half set as training data and Run B with the second half, ultimately retrieving a forecasting error for the full range [59]. Such a modification is trivial to implement and should be considered in future work.

Collation period

The collation period is a critical parameter in the dataset tool, which should be set to balance the following aspects:

- **Signal detail** — As shown in Section 3.1.1, telemetry channels have varying rates of change, levels of discretisation and periodicities, with voltages rapidly changing each minute and some temperatures slowly changing over minutes or even hours. For voltages, having a small collation period near the ideal sample rate would be best as a large amount of information would be lost otherwise. For channels with slow movement and high discretisation, a higher collation period yields almost no difference.
- **Forecasting model train and run times** — As an example, raising the collation period of a channel with a three-second sample rate to one minute lowers the final number of data points by 20 times. Low runtimes are ideal when experimenting with other settings and for model tuning.
- **Forecasting model context window** — Autoregressive forecasters have a look-back parameter which controls how many time steps of past data are presented to the model to form a prediction. With a higher collation period, a model has a larger context window for the same number of look-back.

The combination of a higher collation period and the 'last' aggregation method discussed in Section 5.1.1 reduces the number of spike-type point anomalies present in a dataset, which is fortunately not a significant loss as these anomalies were deemed as uninteresting by instrument engineers anyway (see Section 4.2.3). Early experimentation on temperature channels found no other detrimental effects when using collation periods of one minute and as such this value was used as default for the remainder of the thesis.

Choosing covariates

Covariates may be selected to improve the forecasting performance on a dataset. By default, the orbital and eclipse data introduced in Section 5.1.2 as well as the primary and auxiliary modes of the relevant instrument are added. Often, other relevant telemetry channels are added as covariates to ensure the model is able to keep up with changing operational regimes. The selection is performed carefully, as the use of 'duplicate channels' that are too similar to the target will result in data leaks and should instead be used as additional target channels.

Multiple target channels can be forecasted at the same time, although those selected are almost always duplicate channels or those that have very similar underlying physics. The loss function used to train the models does not account for the individual performance of channels and will thus perform worse if one or more are harder to forecast.

Initially, all non-duplicate channels were selected for forecasting but later experimentation found that removing those that are irrelevant can greatly improve performance while reducing computational time. The final choice of covariates is found by iteration: channels are removed in batches until model performance starts decreasing significantly. This method is extremely time consuming and would ideally be replaced by applying existing knowledge about the physical links between the channels to improve selection. The stability of autoregression with and without covariates is tested in Section 5.2.3

5.2.2. Forecasting setup

Figure 5.5 provides a simplified overview of the forecasting setup (itself a slightly more detailed version of Figure 4.1). Each model in Darts contains a number of hyperparameters which may be used to optimise the performance of a model. To simplify the whole training process, a robust training setup was created which allows any Darts model to be created, trained and stored using a unified model configuration file containing all hyperparameters. The seed of all relevant random number generators is also set to be constant across a comparison.

Model training algorithms (e.g. gradient-descent) utilise a loss function to perform their optimisation, with the Mean Square Error (MSE) most commonly applied by practitioners. The Mean Absolute Error (MAE) was chosen for this project instead because it is less sensitive to outliers and thus better suited to learning the healthy behaviour of a model. Once sample weights include a larger number of rare nominal events and anomalies, MSE may again be considered by comparing the two loss functions in a benchmark comparison. Sample weights, introduced in Section 5.1.5 are provided to the model in the training step. At points where there is missing data or an eclipse, the sample weight of zero causes the loss at those points to also be set to zero, preventing their them from being learned.

During forecasting, the model is provided with a portion of true target data to start the autoregression: 20000 time steps are used before the autoregression starts using forecasted data. These 'warm-up' time steps are not used when measuring the performance of a model.

Once trained, model performance is measured with a number of metrics such as the MAE, MSE and the Symmetric Mean Absolute Percentage Error (SMAPE) which are all aggregate derivations of the forecasting error. The coefficient of determination (R2 score), is another popular measure, which ranges in value from 1 (perfect), 0 (as good as predicting the mean) and any negative value (worse than predicting the mean). It can also be used to compare forecasters that are run on different scales. In addition to applying metrics on the full time scale, they are also calculated per orbit to provide for a more robust perspective on the training data. As the sample weights are not used when calculating metrics, some sections of the data may contain large outliers which may skew the results.

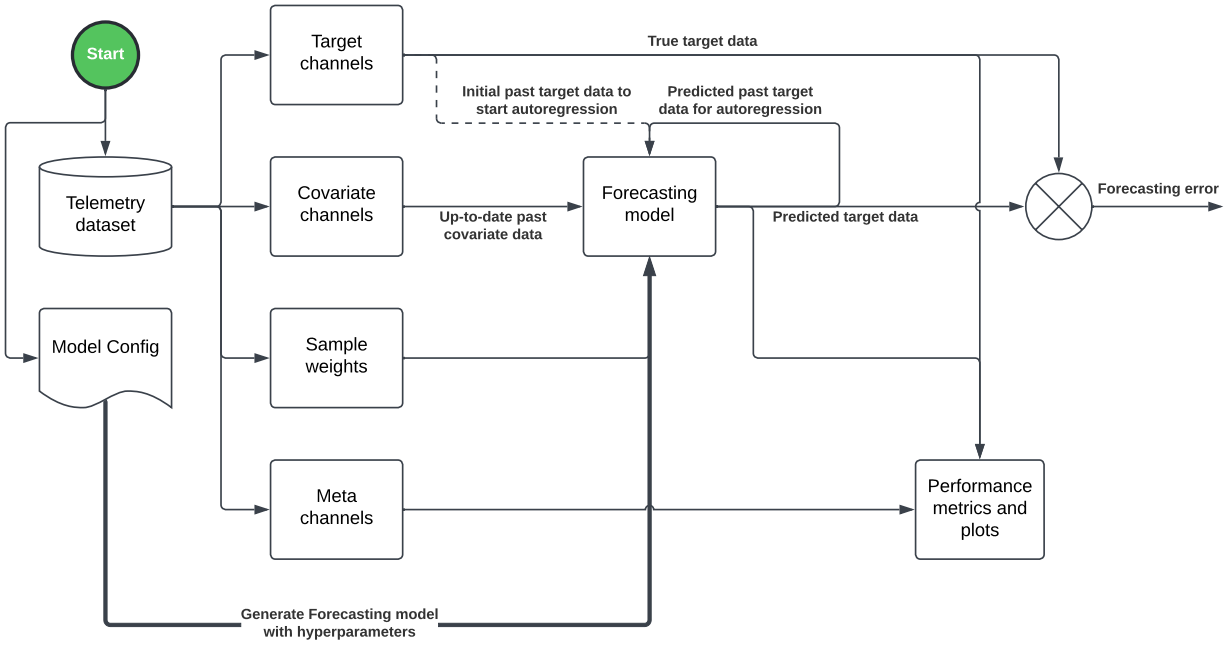


Figure 5.5: Simplified overview of the forecasting setup, including sample weights and metrics.

5.2.3. Training, comparing and selecting forecasting models

In this section various models and dataset parameters are tested to show their performance. Because of their widespread popularity, forecasting was primarily done using various iterations of XGBoost and Long short-term memory (LSTM)-based models. To provide a balanced overview, a large number of compatible models available in Darts are compared to check how they perform on XMM data. For each benchmark model listed in Table 5.2, all hyperparameters are set to default except for the batch size (set to 32), number of epochs (set to 10) and look-back parameter (set to 12 time steps) (*lags* or *input_chunk_length* depending on the model). The XGBoost implementation in Darts uniquely allows for models to be created with spaced-out look-backs (e.g. 1, 5, 15, 60 instead of 1, 2, 3, 4, ...) and an additional model is added to compare the efficacy of this property.

Table 5.2: Benchmark forecasting models. Unless stated otherwise, all models use default settings (as set in Darts version 0.32) with 10 epochs, a batch size of 32 and a look-back of 12 time steps. XGBoost models do not use gradient descent and as such do not have the parameters batch size and epochs.

ID	Darts Model Name	Architecture	Unique parameters	Original
BM_dlinear	DLinearModel	Transformer		[74]
BM_lstm	BlockRNNModel	Long Short-Term Memory	model: ”LSTM”	[75]
BM_nbeats	NBEATSModel	Feed-Forward Neural Net		[76]
BM_nhits	NHiTSModel	Feed-Forward Neural Net		[77]
BM_nlinear	NLinearModel	Transformer		[74]
BM_tcn	TCNModel	Convolutional Neural Net		[78]
BM_tide	TiDEModel	Encoder/Decoder		[79]
BM_transformer	TransformerModel	Transformer		[80]
BM_tsmixer	TSMixerModel	Feed-Forward Neural Net		[81]
BM_xgb	XGBModel	Gradient Boosting		[48]
BM_xgb2	XGBModel	Gradient Boosting	lags (lookback): [-1, -2, -3, -4, -5, -15, -30, -60]	[48]

These models are run on variations of three different benchmarks to compare the models, check their stability over long sets of time and test the data tool scaling options and collation parameter. The benchmarks, elaborated further below, use the channels shown in Figure 5.6.

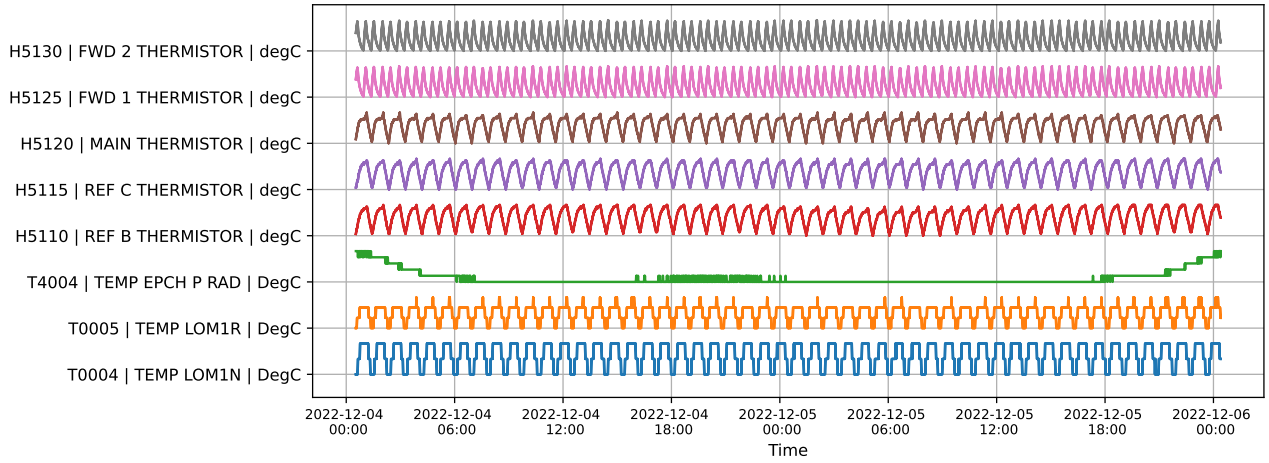


Figure 5.6: Channels used to benchmark the performance of the various models and dataset tool parameters.

Comparison 1: general model comparison

The first benchmark with the parameters shown in Table 5.3, tests all forecasters on a relatively simple problem: a short forecast on easy to predict, stationary channels with a number of covariates. Figure 5.7 shows the results: models *BM_lstm*, *BM_nbeats*, *BM_nhits*, *BM_tide* and *BM_transformer* all show similar performance, with *BM_lstm* performing best overall. In the context of this benchmark data, which is scaled between 0 and 1, an MAE of 0.05 indicates a 5% error. Creating a dummy model that constantly predicts the median gives an MAE of 0.29. The presence of anomalous segments in the data mean no model can produce a perfect reproduction. Visual inspection is used to judge the reproduction of the true signal, and metrics can only be used for relative comparison. The XGBoost model using spaced lags (*BM_xgb2*) appears to perform slightly worse than the version that uses the past 12 lags, although it is possible that newer data is more useful to the model for this specific case.

Figure 5.8 shows training and forecasting times, with the XGB models performing best and transformer-based *BM_nbeats* and *BM_transformer* taking the longest to run. The runtime performance of XGB does not carry over to larger datasets, occasionally taking hours to predict the full dataset, as shown in Figure 5.9. Deep learning-based models scaled much better. Models with short run/training times are preferred, as they speed up manual and automatic tuning.

Table 5.3: Benchmark 1, set to be a relatively easy forecasting problem. All OM temperatures are provided as covariate channels.

Benchmark 1: OM_T0004_T0005_benchmark	
train range	2021-1-1 to 2022-1-1
validation range	2020-6-1 to 2021-1-1
test range	2022-1-1 to 2023-1-1
target channels:	T0004, T0005
covariate channels:	H5110, H5115, H5120, H5125, H5130, H5135, P1035, T2009, T2013, T2017, T2028, T2041, P1135
engineered covariates:	eclipse, distance_earth, distance_sun, beta_angle
mode_channels:	H5395, H5240, HD013
collation period:	60 seconds
col. agg. method:	last
global seed:	3
scaling:	minmax
sample weights:	on

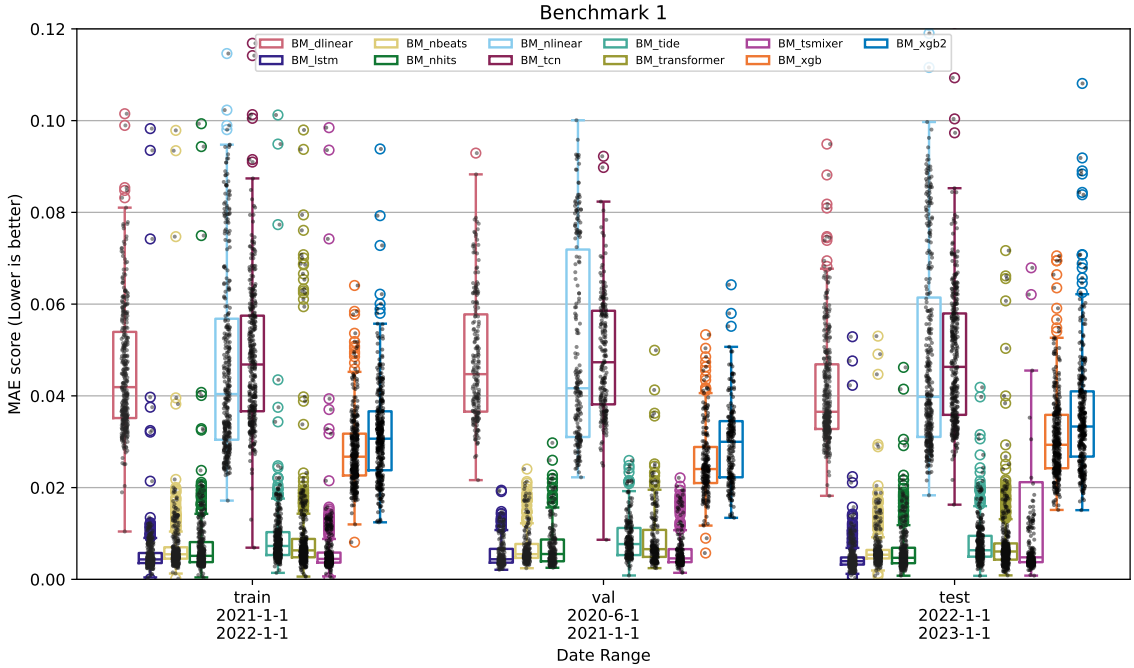


Figure 5.7: Model performance for benchmark 1. Metrics are calculated per channel and per orbit. Channels T0004 and T0005 are very similar and were deemed suitable to be grouped together for comparison.

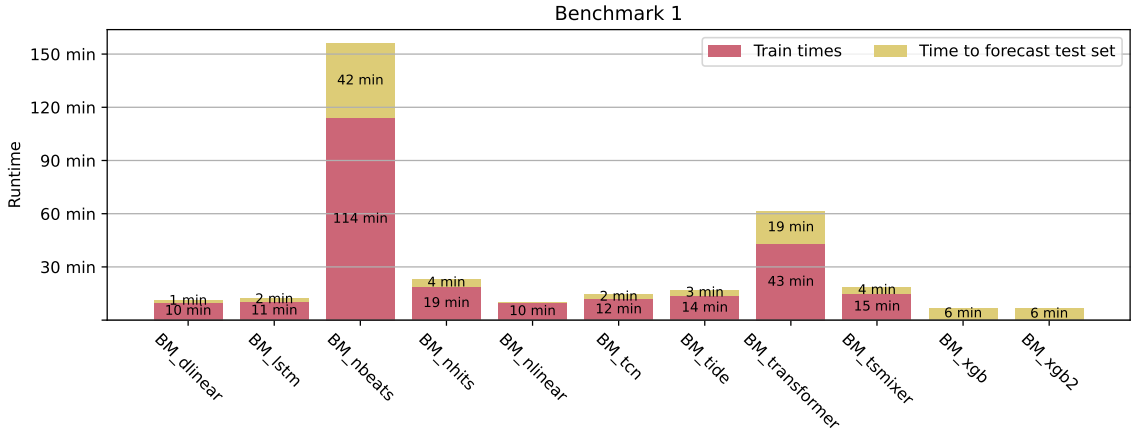


Figure 5.8: Training and forecasting times for each model in benchmark 1.

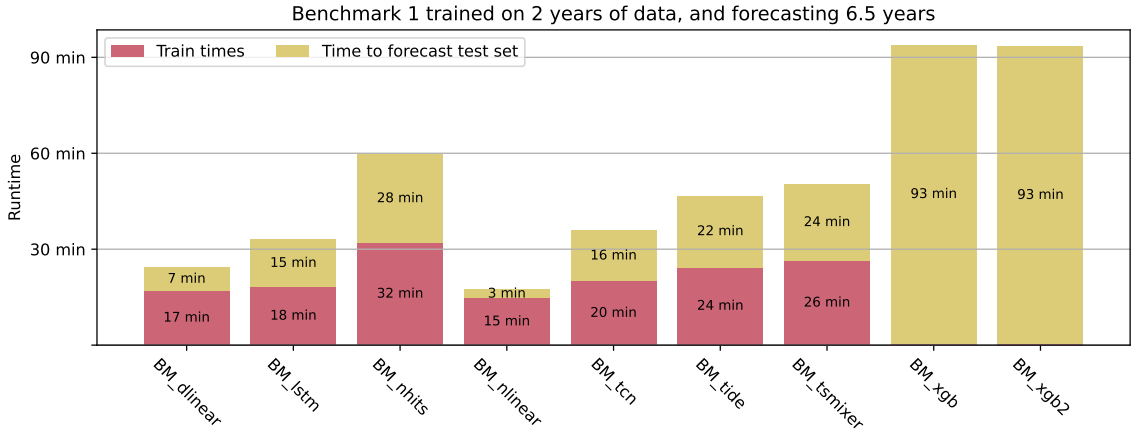


Figure 5.9: Training and forecasting times for select models for a larger dataset with equivalent settings to benchmark 1.

Comparison 2: data scaling

Section 5.1.3 presented three scalers, *MinMaxScaler*, *RobustScaler* and *StandardScaler* which are compared in Figure 5.10. The previous benchmark dataset is ran twice (seeds 1 and 2) for each scaler using model *BM_lstm*. While the performance is not significantly different across scalers, *RobustScaler* does provide consistently better results across over the full collection orbits, indicating that it can be used for marginal benefit. This also matches literature [82]. That being said, the channels tested (T0004 and T0005) have a somewhat Gaussian distribution and a similar performance increase may not be possible on telemetry with a different distribution. As this result was found quite late into the thesis, most of the final results are still generated with the MinMaxScaler.

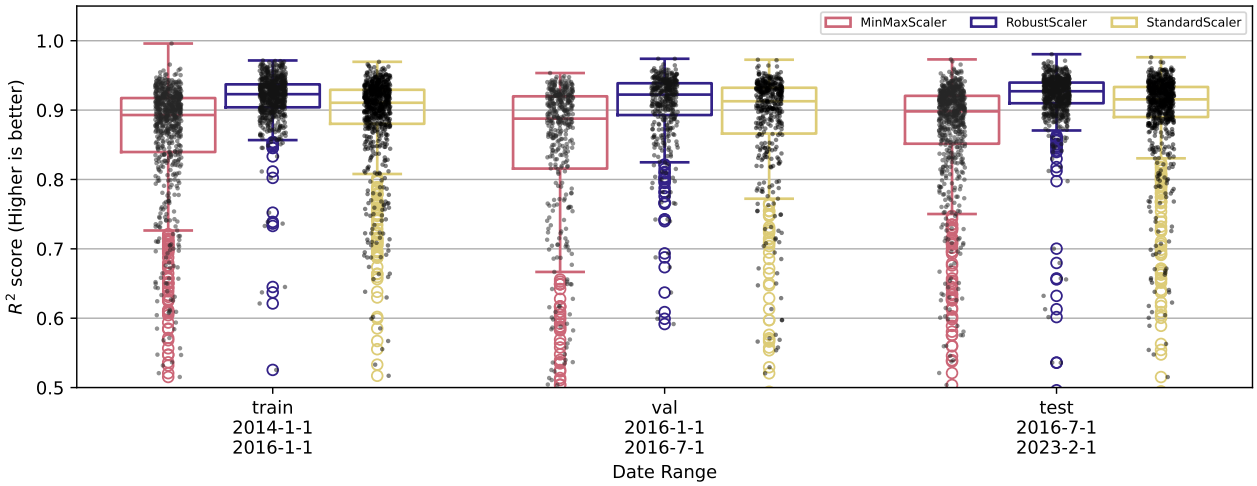


Figure 5.10: Comparison of scalers. Metric R2 is used to allow for comparison in different scales.

Comparison 3: Use of covariates and forecasting quality over time

To test the robustness of a forecaster in autoregression for a longer time, Comparison 3 tests the ability of a forecaster to robustly perform autoregression for a longer period of time. Table 5.4 lists the parameters for benchmark 2, which covers 5 thermistor channels of OM over the full 9 years of data. Two versions of the dataset are run (again with *BM_lstm*), one without covariates and the other with two power related covariates, with the idea that the power covariates could serve as crutches to ensure stable autoregression. Figure 5.11 yields a surprising result: the versions without covariates appear to perform better. Figure 5.12 shows each model performing well until a first eclipse season is reached. At that point, both models on seed 4 fail to return to normal autoregression. The bad performance here can be attributed to a failed initialisation or conversion during training. On seed 1, the version with covariates fails at a later eclipse before recovering after some time. The reason for this jump was not analysed but it might potentially be caused by an over-reliance on the P1135 channel, even though the modes provide sufficient information for a successful forecast. Nevertheless, earlier experimentation found that some channels cannot rely on just the modes and as such, the addition of covariates should be considered on a case by case basis.

Table 5.4: Benchmark 2 used to check the robustness of autoregression over long periods of time in comparison 3. Two options for covariate channels are used.

Benchmark 2 :OM_Thermistor_benchmark	
train range	2014-1-1 to 2016-1-1
validation range	2016-1-1 to 2016-7-1
test range	2016-7-1 to 2023-2-1
target channels:	H5110, H5115, H5120, H5125, H5130
covariate channels:	None or P1135, P1035
engineered covariates:	eclipse, distance_earth, distance_sun, beta_angle
mode_channels:	H5395, H5240, HD013
collation period:	60 seconds
col. agg. method:	last
global seed:	1, 2, 4
scaling:	minmax
sample weights:	on

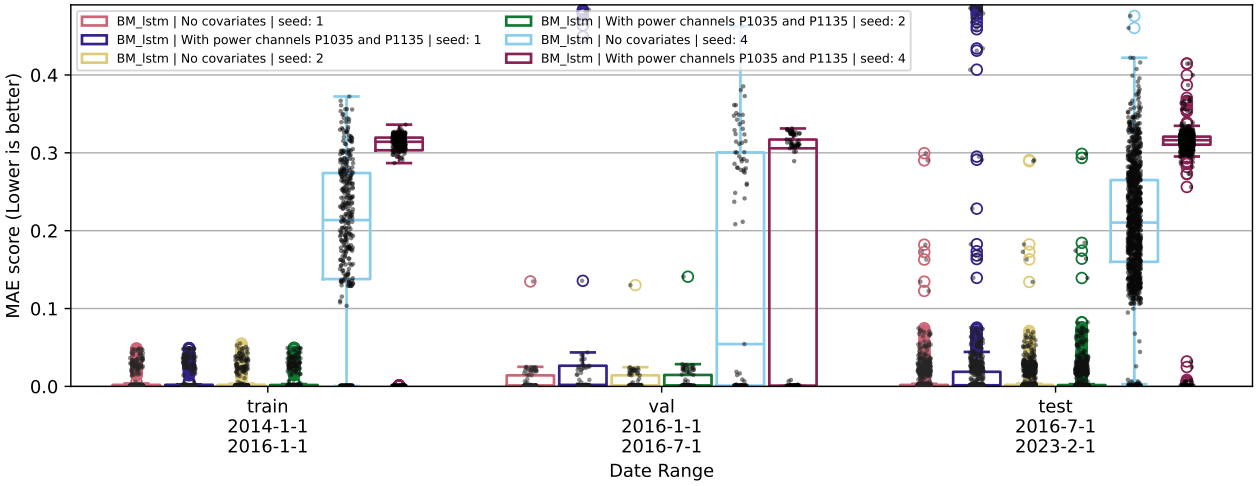


Figure 5.11: MAE results for comparison 3. Three runs show significantly worse performance.

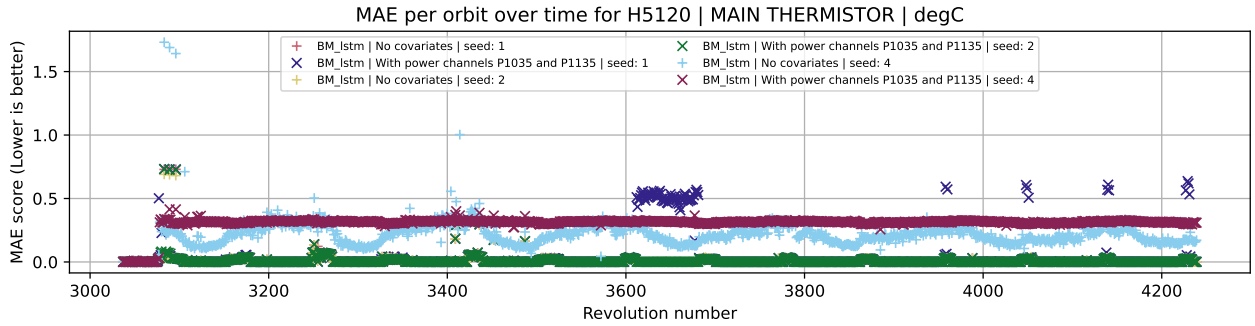


Figure 5.12: MAE per orbit over time for the various runs.

Comparison 4: Different collation periods

The collation period (discussed in Sections 5.1.1 and 5.2.1) is a key parameter in the dataset tool. This comparison tests its effect on the forecasting model's context window. A new benchmark dataset, shown in Table 5.5, focusses on T4004, the radiator temperature for PN which only displays a large fluctuation near the periapse before slowly lowering to a constant temperature for the rest of the orbit. The channel proved difficult to forecast on the standard 1 minute collation period (12 min look-back) and is therefore compared to a longer period of 15 minutes (3 hour look-back) and 30 minutes (6 hour look-back) in Figure 5.13. The results show a clear benefit in using a longer collation period to provide a wider context window. As an alternative to a larger collation period, a larger look-back can also be used, at the cost of significant increases in runtime. An XGBoost forecaster can

be configured to use wider spacing in its look-back to achieve similar results (not tested here). It can also be seen that *BM_xgb* generally performs better on this problem than *BM_lstm* indicating that the best performing model varies per channel.

Table 5.5: Benchmark dataset 3 to compare various collation periods. Sample weights are disabled as a bug in the XGBoost implementation in Darts did not support them for a single target channel.

Benchmark 3 :PN_T4004_benchmark	
train range	2021-1-1 to 2022-1-1
validation range	2020-6-1 to 2021-1-1
test range	2022-1-1 to 2023-1-1
target channels:	T4004
covariate channels:	F1122, F1198, F1199, F1201, F1189, F1190, F1192, F1257, F1258, F1259, F1260, F1191, 'F1193
engineered covariates:	eclipse, distance_earth, distance_sun, beta_angle
mode_channels:	FD126, FD130,
collation period:	1 minute / 15 minutes / 30 minutes
col. agg. method:	last
global seed:	1
scaling:	minmax
sample weights:	off

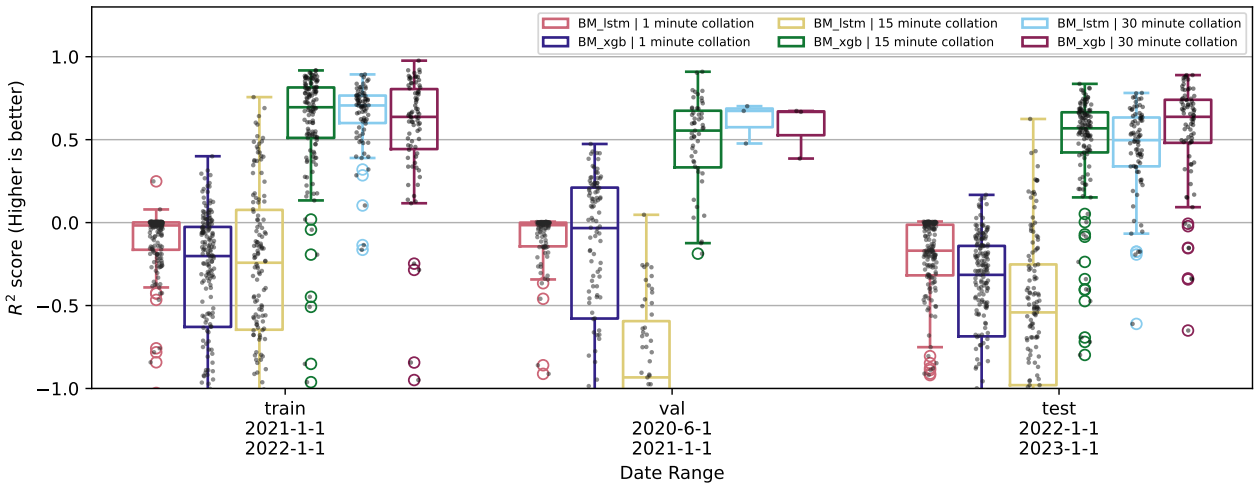


Figure 5.13: Comparison for collation periods for comparison 4. The R2 score is chosen as a metric here as T4004 is flat for most of the orbit (see Figure 5.6).

Key takeaways

The comparisons shown in this section cover a very small part of the picture. The large number of parameters made available by the dataset tool as well as the large number of models and their hyperparamters, each with potential yields in performance result in an overwhelming amount of choices. As an example, each model may perform better when tuned to accommodate the data presented. Various versions of XGB were found to outperform LSTM models for certain target channels. While not conclusively proven in this thesis, it appears that no one model performs best for all channels, which echos the findings of many publications in Section 2.2.3 on a much smaller scale.

The abundance of parameters does not matter much in the well-behaving data shown in comparisons 1, 2 and 3, but are sometimes paralysing in cases where an initial good forecast is hard to find. This issue was partially solved by running multiple variations of multiple models on each new target channel. Usually, one family would then perform best.

Throughout the project, LSTM-based models were found to be simple, fast, and broadly compatible with many target channels and they are used to generate the final results in Chapter 6.

5.3. Anomaly scoring and detection

A suitable forecasting model can be run to find a forecasting error, which can then be processed in various ways to find an anomaly score. The most common scoring method simply uses the absolute forecasting error. A quantile threshold can then be applied to mark anomaly detections and to avoid small forecasting errors. Figure 5.14 shows an example anomaly detected using an absolute error anomaly score and quantile threshold of 97%. In experimentation, the use of the raw forecasting error yielded many point detections caused by small mistakes in forecasting. Say a telemetry channel makes a large jump, if a forecaster predicts this jump one time step too early or to late, the forecasting error is left with a single data point with very high error which may be caught by the thresholding as an unwanted detection. Additionally, a forecasted prediction may intersect with the true value, leading to multiple small detections instead of a single large one (also seen in Figure 5.14). Literature has shown that processing the absolute error with an anomaly scoring function may significantly improve anomaly detection performance [52]. A number of published forecasting-based algorithms even have their own built in scoring method [13][83]. In Section 5.3.1 a number of scorers are explored and compared, demonstrating their consequential effect on the detection of anomalies. The limitations and efficacy of the scoring method is discussed in Section 5.3.2.

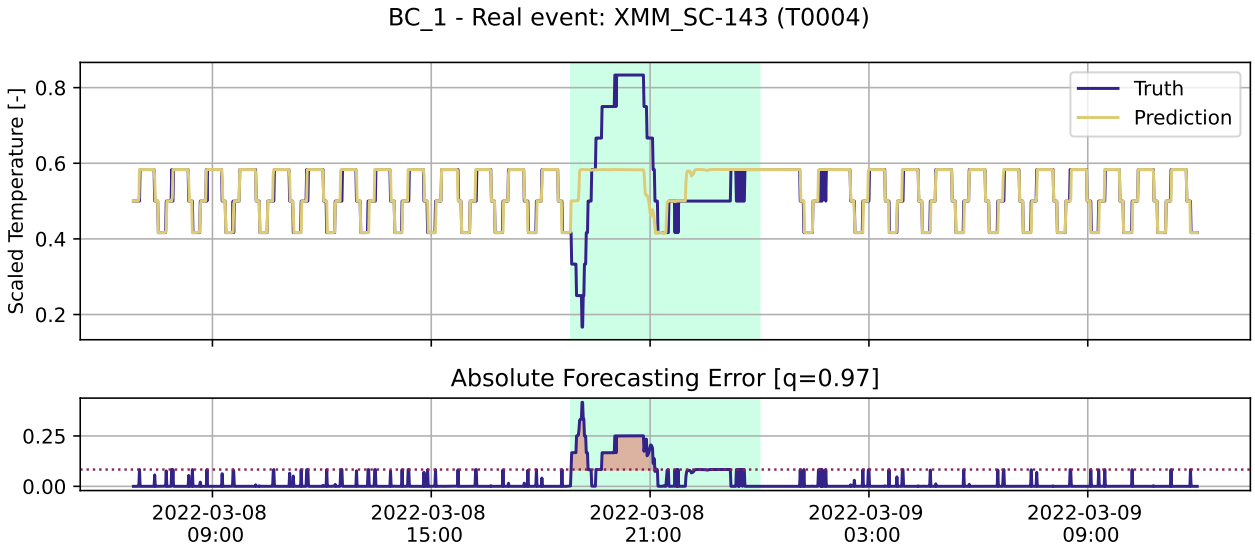


Figure 5.14: An example detection with the absolute error as the anomaly score and a quantile detection threshold of $q = 0.97$ (top 3% highest anomaly scores). The green area is the visually perceived anomaly, the red area shows the area that is detected by the scorer + detection threshold combination. The event would in this case be caught in two separate detections, as the forecasting error goes to zero in the middle. The use of a quantile threshold prevent the detection of the many small forecasting errors surrounding the event.

5.3.1. Comparing scorers

Once ground truth anomalies were provided in the form of the anomaly reports shown in Section 3.2, the following steps were used to select a scorer:

1. Forecasting was performed on channels with known anomalies. The resulting forecasting error was used as an anomaly score directly and thresholding was used to find an initial set of anomaly candidates.
2. Anomaly candidates were screened to find a set of 'benchmark candidates' that seemed most unique and interesting. Not all candidates were clear or confirmed anomalies. Some clear forecasting errors were included to check how the scorers would behave under interesting conditions, others were included as examples of events which should not be detected.
3. The start and end time of the benchmark candidates was manually adjusted to cover what is visually perceived as an anomaly.
4. A number of scorers are selected and applied to the forecasting error, then detections across all scorers are made using the same quantile.
5. For each benchmark candidate, the marked start and end times of the candidate are visually compared to what would have been detected by a scorer. Figure 5.14 shows the visual process for a single scorer.

Initial success was found on OM temperature channels T0004 and T0005 with a diverse sample set of benchmark candidates shown in Figure 5.15. The relevant processed telemetry data, resulting forecast and benchmark candidates shown in the remainder of the section were stored to provide consistent analysis, but are oth-

erwise outdated and may appear different in other sections. Two dedicated scorers from literature, the *Gauss Static* and *Gauss Dynamic* scorers [52] were reimplemented for comparison. Unsupervised TSAD algorithms have also successfully been used as scoring functions [12]. Darts allows for a number of unsupervised TSAD methods to be applied on the forecasting error to find an anomaly score. The unsupervised distance-based algorithms Histogram-Based Outlier Score (HBOS) and K-means, as well as Isolation Forest, all previously described in Section 2.2.2, are also applied with varying parameters.

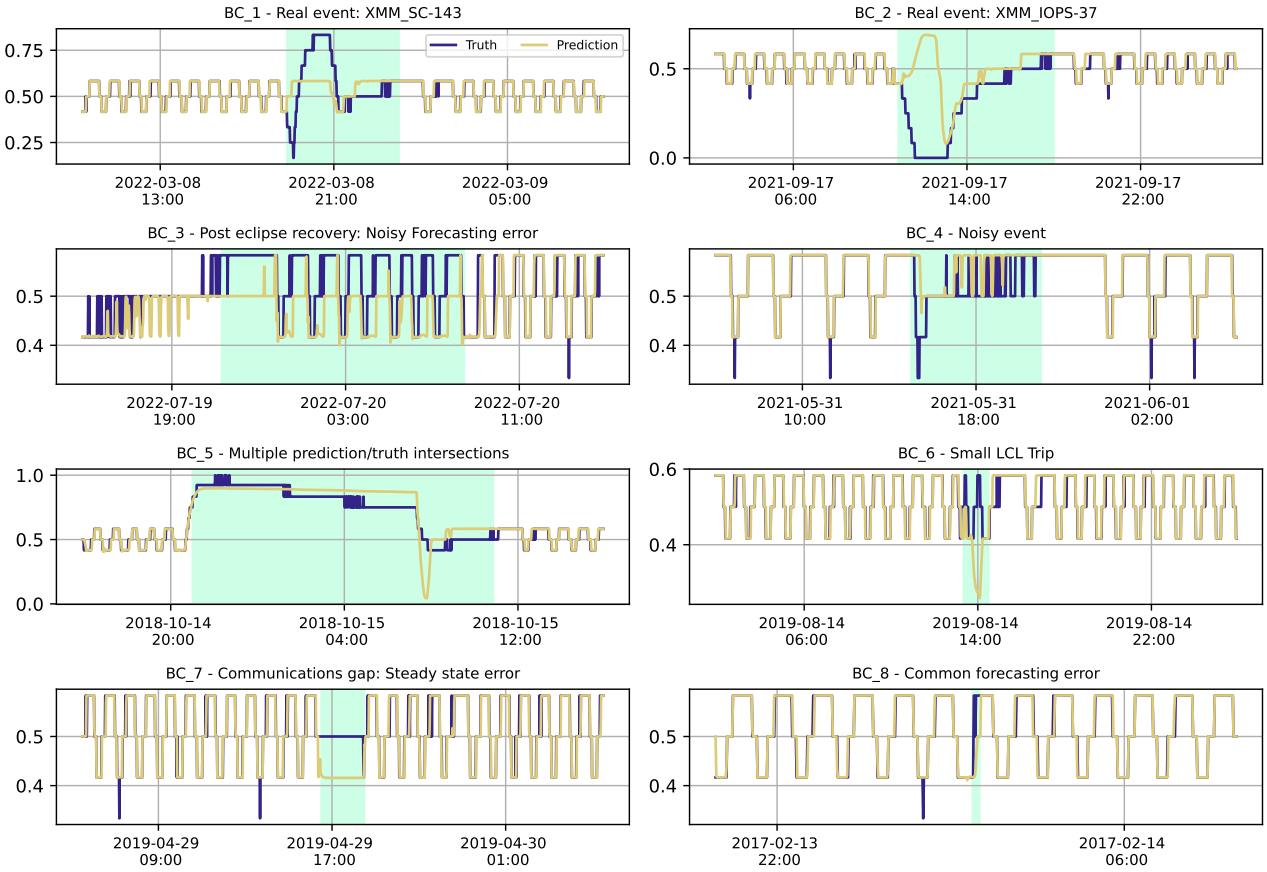


Figure 5.15: Example benchmark anomaly candidates used for selecting anomaly scorers. Marked anomaly is highlighted in green.

Each unsupervised model has a unique method of fitting to the data. In K-means clustering, the complete forecasting error sequence of length L is divided using a moving window method of stride 1, creating $L-W+1$ window vectors of size W which are then grouped into K clusters. Because the method is performed per channel, each ingested vector is one-dimensional with length W. Vectors are then compared by calculating the difference at each index of the vector and combined into a euclidean distance, which is used as the clustering metric. Effectively, the vectors are clustered by their shape. For HBOS, the vectors are binned forming a histogram with the vectors in the short bins marked as the most anomalous. The core Isolation Forest method is explained more extensively in Section 2.2.2. Here, anomalous vectors are expected to be rare and easy to isolate. Once fit, these models can then be applied to score the window vectors. At each point in time, the score of each window which includes this data point is aggregated to achieve a final anomaly score for each point in time. The process is repeated independently for each target channel.

The formula for the static Gauss score G_s at time t is shown in Equation (5.5), with Φ the cumulative distribution function for the standard normal distribution, E the forecasting error, μ_E the mean of the forecasting error and σ_E the standard deviation of the error. An infinitesimal ϵ is added to prevent discontinuities. The dynamic Gauss score G_d, is similar to the static version with the mean and standard deviation calculated over a window W preceding the current time t, instead of the whole timespan. Note that the window used for the dynamic Gauss score is different from that of the unsupervised algorithms and is used to provide seasonal context to the scorer. A third Gauss score with a smoothing function is not replicated as its performance did not appear to have significant improvement over the unsmoothened dynamic Gauss score G_d [52]. Instead, some iterations of the Gauss score with a moving average-smoothened forecasting error were tested.

$$G_s(t) = -\log\left(1 - \Phi\left(\frac{E(t) - \mu_E}{\sigma_E}\right) + \epsilon\right) \tag{5.5}$$

For each scorer, various parameter configurations, most notably varying in window size, were tested and compared. A sample of the comparisons is shown in Figure 5.16. The full comparison yielded the following observations and interpretations:

1. *Subsequences where the prediction deviates significantly from the measured signal for extended periods (e.g. BC_1, BC_2 and BC_5 in Figure 5.15)* — All scoring methods are able to detect these anomalies, however Isolation Forest and HBOS appear to cover the perceived anomaly best. Although these anomalies are usually also detected by the raw absolute error, HBOS, moving average smoothing and Isolation Forest do a better job at capturing the full anomaly.
2. *Noisy subsequences where there are extended periods of small forecasting error (e.g. BC_3 and BC_4 in Figure 5.15)* — Most scorers tend to miss these anomalies as their absolute error may be really small. *BC_4* proved particularly challenging. Scorers are able to detect the anomalies but their detection covers only a small section of the perceived anomaly.
3. *Subsequences where the there is a steady-state error between the prediction and measured signal (e.g. BC_7 in Figure 5.15)* — These are detected best by Isolation Forest.
4. *Brief spikes (e.g. BC_6 in Figure 5.15)* — These are best caught by the the raw absolute error, the gauss scorer or variants of K-Means. Scorers with a high window size smoothen out the spikes. As per Section 4.2.3, these anomalies are less interesting for this project as anomalies can already be caught by out-of-limits systems.
5. *Brief, clear forecasting errors (e.g. BC_8 in Figure 5.15)* — K-means, Gauss and the raw absolute error are prone to catching common forecasting errors. The example presented at the start of this section, where a large jump is predicted one time step too soon or too late is frequently caught by these scorers. The Gauss scorer and absolute error are obviously most susceptible to such events as A possible explanation for K-means is that these types of event are more common and different enough from healthy data to warrant their inclusion in the anomalous cluster. The use of more than 2 clusters and the modification of other parameters may improve the performance of K-means. The fact that these events are relatively common makes them less likely to be marked as anomalous by HBOS and Isolation Forest.
6. *Intersections between the true telemetry signal and the prediction (e.g. BC_5 in Figure 5.15)* — Here larger anomalies are split in two or more detections as the forecasting error becomes zero at intersections between the predicted and measured telemetry signal. Such detections are better captured by scorers with higher window sizes but have not been fully captured by any of the scorers tested. As a workaround, a post-processing step introduced in Section 5.4 merges detections that are very close to each other.

Ultimately no single scorer tested was able to completely cover all benchmark anomaly candidates. The window size is a particularly important parameter, small window sizes are unable to contextualise anomalies with a longer duration and long windows smoothen out shorter anomalies. To cover both, multiple scorers were ultimately used with their detections combined. Some candidates, such as *BC_4* were not adequately covered by any tested scorer. Table 5.6 shows the scorers tested (all with a 97% threshold) and those chosen, with their final quantile thresholds. While 3 combinations of HBOS and Isolation forest were ultimately chosen, it should be noted that their performance is only somewhat better than simpler approaches such as the moving average. The smoothing effect provided by the window parameter of each scorer has a far greater effect than the underlying technique.

Eclipses often cause the most extreme values for a telemetry channel (as shown in Section 3.1.4), resulting in a high forecasting error for a relatively long period of time. While events like eclipses and missing data can be considered rare nominal events which should be annotated in an anomaly benchmark, their presence significantly alters the distribution of the anomaly score, affecting the quantile detection system. Figure 5.17 shows *BC_3* in a broader context, between two eclipse events. At a later state of the project an additional processing step was introduced (shown in Figure 5.18), setting the forecasting error to zero at points where there is an eclipse or missing data, curbing their influence. This step also lead to higher quantile detection thresholds, which is reflected in the final scorer selection in Table 5.6. Further improvements in the detection post-processing discussed in Section 5.4 and implemented after this comparison also alter the ultimate efficacy of the scoring and detection, potentially altering the final choice made in this section. Ideally, the whole scorer analysis provided in this section should have been repeated, potentially yielding a different outcome, but this was not done due to time constraints.

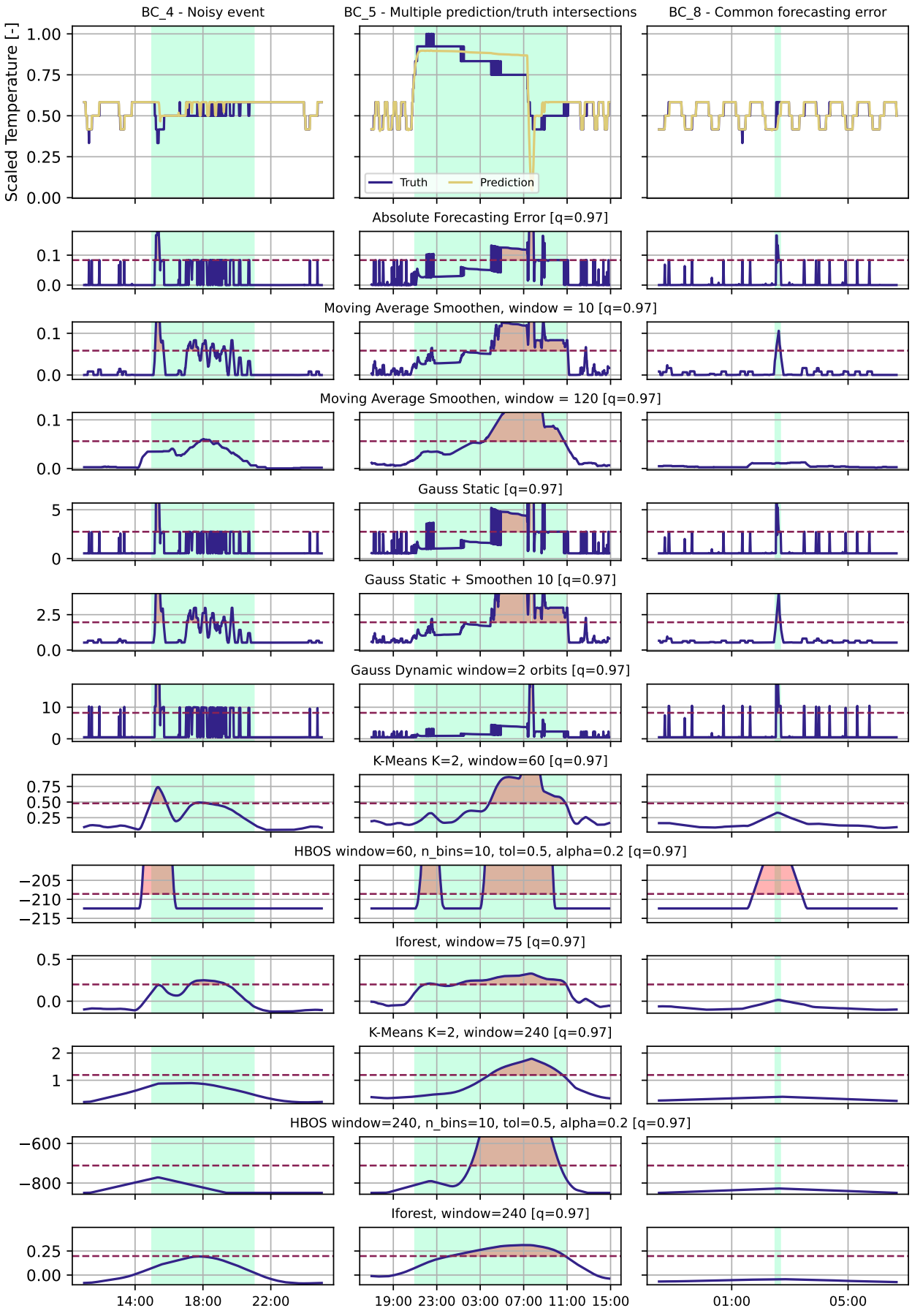


Figure 5.16: Scorer comparison for a sample set of scorers and benchmark candidates. The detected area is shown in red, while the perceived anomaly is shown in green. BC_8 shows an example of a common forecasting error to be avoided. A constant quantile detection threshold of 0.97 is used for all scorers.

Table 5.6: The various scoring function configurations that were teste. Scorers used for final results are highlighted with their respective quantile thresholds. All unlisted settings are set to the defaults of Darts version 0.32 and PyOD version 2.0.2.

Scorer	Configuration	Selected
Raw absolute error	-	
Moving Average	window = 10	
	window = 75	
	window = 120	
Gauss static	-	
	With moving average (window=10) applied to raw error	
Gauss dynamic	window=6000	
	window=12000	
	window=20 orbits (57600)	
Isolation Forest	window = 30	
	window = 60	
	window = 75	Yes (threshold q=0.998)
	window = 120	
	window = 240	Yes (threshold q=0.995)
	window = 600	
K-Means	K = 4, window = 30	
	K = 2, window = 30	
	K = 2, window = 60	
	K = 2, window = 75	
	K = 2, window = 120	
	K = 2, window = 240	
	K = 2, window = 600	
HBOS	window=60, n_bins=10, tol=0.5, alpha=0.2	
	window=70, n_bins=10, tol=0.5, alpha=0.2	
	window=120, n_bins=10, tol=0.5, alpha=0.2	
	window=240, n_bins=10, tol=0.5, alpha=0.2	Yes (threshold q=0.995)
	window=240, n_bins=5, tol=0.5, alpha=0.2	
	window=240, n_bins=15, tol=0.5, alpha=0.2	
	window=240, n_bins=10, tol=0.5, alpha=0.3	
	window=240, n_bins=10, tol=0.5, alpha=0.1	
	window=600, n_bins=5, tol=0.5, alpha=0.2	

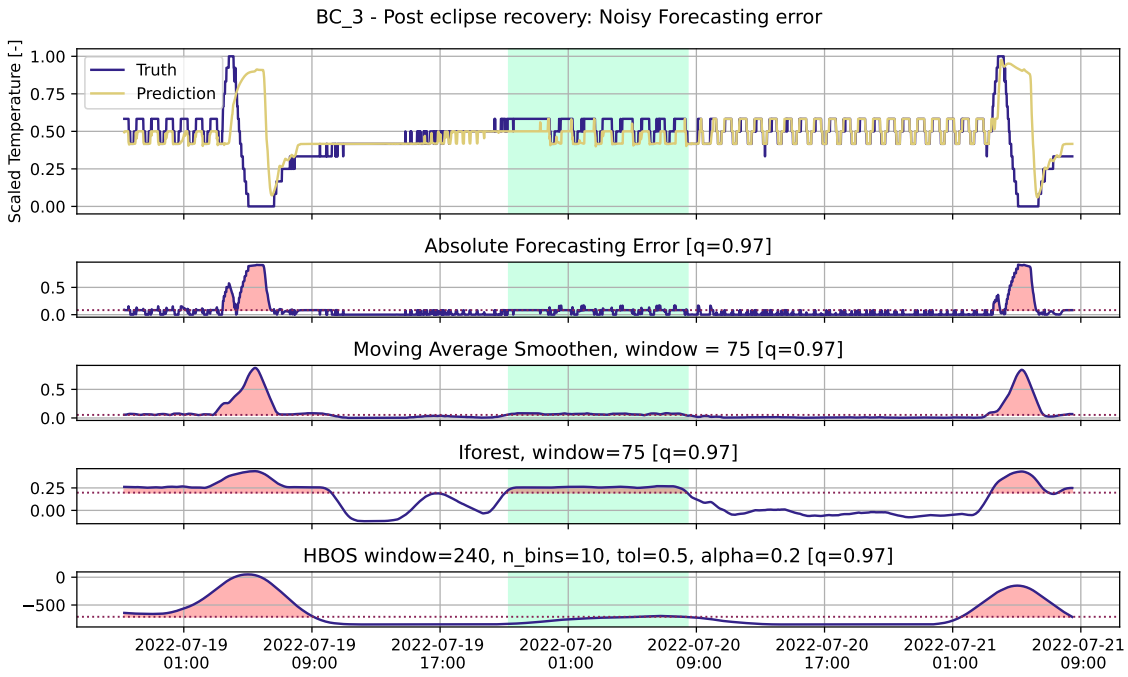


Figure 5.17: An expanded view of BC_3, showing how the two eclipses on each side are more prominently caught by the scorers.

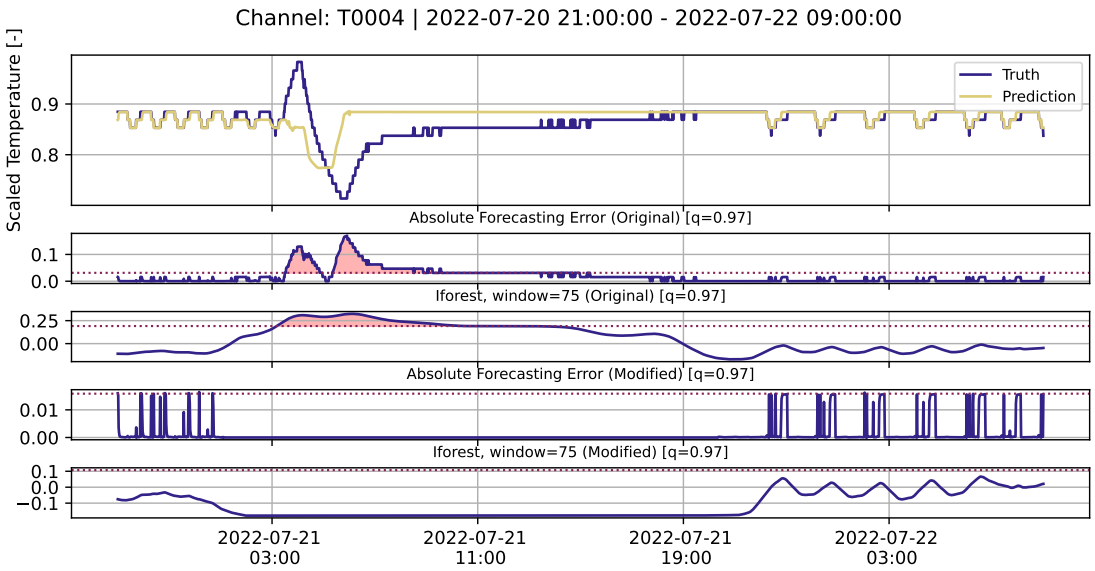


Figure 5.18: Scoring for an eclipse event, comparing the original scoring and the modified version which removes eclipses.

5.3.2. Scorer method limitations

The modification of the raw absolute anomaly score has clear benefit in anomaly detection, reflecting earlier results by Garg et al. (2021) [52] and Gomez et al. (2024) [12]. Even the application of a simple moving average removes the large number of small forecasting mistakes as can be seen in Figure 5.17. While the application of unsupervised anomaly detection algorithms as scoring functions were perceived to yield slightly better results than the simpler moving average and gauss scoring functions, their use comes at the cost of a higher runtime and computational cost.

All of the methods tested were only supplied with the absolute forecasting error, which does not provide the complete context of an anomaly. The value and pattern of a telemetry channel as well as its distribution can be useful in determining whether a section of telemetry data is anomalous or not. Future attempts at using unsupervised models such as HBOS and Isolation Forest may try to include such additional context. While scorers such as Gaussian Mixture Models (GMM) exist that focus specifically on the distribution of the data presented, limited experimentation with these models did not achieve desirable results when supplied with the absolute forecasting error alone. Additionally, the potential for a forecaster to predict perceived anomalous behaviour, thus lowering the forecasting error, has an influence on the detectability of an anomaly. This issue is clearly seen for *BC_5* in Figure 5.16, where the first half of the anomaly is much more difficult to detect even though the telemetry channel reaches its highest value in the dataset at 1.0. The additional context that the telemetry channel reaches its maximum would be a clear reason to mark the event as an anomaly. While the step to include additional context to the anomaly detection process, either as pure statistical values or in the context of physics-based anomaly detection is some steps removed from the work done in this thesis project, and a thesis-sized project in its own right, an eventual implementation could yield excellent results.

The use of the absolute forecasting error as the only input for each of the scorers combined with the use of quantile thresholding introduce a major blind spot. As each anomaly score is ultimately a modification of the absolute error, events with a high forecasting error (*BC_1*, *BC_2* and *BC_5*) are often easily caught, while events with a low, but sustained forecasting error such as *BC_4* are more difficult to detect. While anomalies with large forecasting error are of clear concern to ground operators and instrument engineer they are also the easiest to catch. An automatic system should ideally also excel at detecting these smaller, inconspicuous anomalies, which may be missed by engineers [5]. Unfortunately, this aspect was not completely solved with the scoring system developed in this thesis.

The static quantile scoring method also brings its own issues. In the case where the forecasting is near perfect (e.g. there are no anomalies), the quantile threshold will still produce anomaly detections. In fact, such cases produce an even higher number of detections since every small little error is marked as an anomaly. Filtering detections by their anomaly score is difficult, as smaller detections may sometimes be as interesting as larger ones. In the future, a method should be developed to better account for such cases. This could take inspiration from a dynamic thresholding algorithm, such as the one used in Telemanom [13].

Manually selecting the parameters of the scorer and the subsequent quantile threshold to make detections is a very labour intensive task. The process could be automated in the future once a large set of benchmark

anomalies is available. A number of scorer, parameter and quantile threshold combinations may be initialised and run, each producing detections. These detections may then be compared to the benchmark anomalies using a classification metric like the F1-score or the Area Under the Receiver Operating Characteristic Curve (ROC AUC). A subsequent step could be to then tune or optimise a set of scorers using the F1-score or ROC AUC as a fitness value. Implementation of such systems without a sufficient and varied number benchmark anomalies may yield inadequate results as such a process is highly susceptible to over-fitting.

Ultimately, the scorer comparison was performed on a limited set of candidate anomalies found on two channels. While there is clear benefit to modifying the forecasting error, there is significant room for improvement in the scoring system presented in this section. Additionally, a more robust and analytical comparison of scoring functions on a wider set of telemetry channels and benchmark anomalies is required to reach a more concrete conclusion on the efficacy of each scoring function.

5.4. Anomaly post-processing and cataloguing

The preceding anomaly scoring and detection step yields a detection mask where columns are the target channels and each row is a point in time. Anomaly detections are marked 1 for each anomalous data point and 0 where no anomaly is detected. This format is particularly useful for an eventual anomaly benchmark as it can be directly used to compare the performance of various TSAD algorithms tested. Nevertheless, anomalies can be analysed to find a number of additional metrics which cannot be stored in a mask format, such as the dimensionality, locality and length metrics used by ESA-ADB. As an alternative, anomalies can be stored in a tabular format, where each row represents an anomaly. The most basic format would include a start time, end time and channel name for as columns for each anomaly. Additional columns could include any required metrics and a unique identifier such that each anomaly may be referenced individually.

ESA-ADB use a tabular format that is expanded into two tables in the form of a relational database. A first table provides an event per row, with the start time, end time and channel. Events may happen concurrently across different channels and concurrent events across all channels are provided with a single ID. A second table then provides additional context and metrics per ID. For this project, a simpler method has been developed to make the data directly accessible. To start, a unique id is provided to each event per channel and concurrent events are collectively provided a group number. Metrics are then provided for each event per channel, resulting in a single table with all the information. The table format is more accessible to humans, allowing for easy manual adjustment and filtering. As anomaly start and times may require manual modification, it is useful provide translation back to the mask format. This functionality has not implemented in this thesis but can easily be replicated from similar processing steps used for the eclipses and out-of-limits data. The storage of anomalies in tabular format also allows for finer start and end times which can be rounded to the chosen collation time when converted to a mask.

In addition to the conversion from a mask to a basic tabular format, the following additional processing steps are performed to achieve a final catalogue in tabular format:

- **Merging detections across different scorers** — As discussed in Section 5.3, multiple scorers were required to cover the wide range of different anomalies present in the data and each scorer yields its own detection mask which is then converted to a tabular format. In this first post-processing step, the collection of tabular detections is merged into a single table. For each channel, overlapping detections across the different scorers are merged into a single detection. Figure 5.19 shows this processing step combined with the next processing step.
- **Merging nearby detections** — As discussed in Section 5.3.1, limitations of the scoring system implemented for this thesis cause a single anomaly to sometimes be captured in more than one detections. This is undesirable as it produces additional detections to be investigated. In this processing step, nearby anomalies with a space of less than four hours between them are grouped into a single anomaly. The 4 hour buffer is adjustable but was found suitable in experimentation. Anomalies are usually sparse and interesting anomalies often required a larger buffer to be captured. Figure 5.19 shows this processing step combined with the previous processing step.
- **Grouping concurrent detections across telemetry channels** — In this project, forecasting and anomaly detection has been performed on multiple target channels concurrently. To simplify manual analysis, detections that occur concurrently across channels are collected into groups such that they may be analysed together.
- **Adding metrics to the detections** — To aid in analysis, detections are provided with a number of categories and metrics. The In addition to an ID, the start time, end time and channel, the following metrics are assigned to each detection, here described by their name in the toolset:

  - *multivariate* — Marked true if there is any concurrent detection in another channel. This is the same as dimensionality parameter used in ESA-ADB.
  - *duration* — The time duration of the detection, calculated from its start and end times. ESA-ADB marks the length of an anomaly as a point anomaly if the length of the event is 3 data points or less, and marks it as a subsequence anomaly otherwise. Because such definitions may be perceived as arbitrary, the numerical time duration is provided instead.
  - *has_nan* — Marked true if there are any missing data points within the bounds of the detection, as marked by the missing data mask discussed in Section 5.1.1.
  - *has_ool* — Marked true if any point in time with the detection overlaps with an Out-Of-Limits (OOL) event.
  - *has_eclipse* — Marked true if any points in time within the detection overlaps with an eclipse event.
  - *origin* — Provides a list of all scorers that formed the detection.
  - *revno* — Provides a list of revolution numbers that this detection spans. The revolution is commonly used by instrument engineers and ground operators to describe anomalies.
  - *mean_anomaly_score* and *max_anomaly_score* — Provides the mean / max value of the anomaly score within the bounds of a detection. This is calculated at the initial conversion from mask to table and is lost when combining tables for multiple scorers. Unavailable in the final version of the post-processing step, but may be reimplemented as a list similar to the *origin* metric in the future.

- **Clustering anomaly detections (Experimental)** — In an attempt to further simplify analysis, similar anomalies were grouped into clusters. The method is inspired by a window-based anomaly detection method applied by Ruszczak et al. (2023) [57] to find anomalies in telemetry data of the OPS-SAT satellite. In this case, a K-Means clustering model is provided the detect metrics above as well as some statistical information such as the mean and variance of the telemetry data and anomaly score in and around the detected segment. The method yielded promising results but was not perfected due to time constraints. While similar anomalies were sometimes clustered together, the method occasionally failed to group the same event across channels, even when both channels are similar in nominal behaviour (e.g. T0004 and T0005) and showed a similar behaviour in the detection. In the future, the method may be improved to further reduce manual analysis time.
- **Filtering by metric (Abandoned)** — The metrics introduced a few points earlier can be used to filter out unwanted anomalies. Over the course of the project two types of filters were considered but ultimately abandoned:
  - Filtering out events with *has_nan*, *has_eclipse* and *has_ool* — These events are known anomalies and rare nominal events and can be added back more precisely in a subsequent step. As such events are frequent and commonly detected, thus adding significant clutter when analysing anomaly detection output, they were originally filtered out of detection tables. An undesireable consequence of such a filtering step is that large anomalies that may briefly overlap with eclipses or cause missing data are then cut. Thus, this filtering step was later abandoned. A preprocessing step for the scoring and detection system discussed in Section 5.3.1 ultimately meant that areas with eclipses and missing data points were unscored anyway, almost completely removing the clutter they introduced.
  - Filtering out events with low mean anomaly score — This filtering step was used very early on but was quickly abandoned. As discussed in Section 5.3.2, the anomaly score is closely tied to the forecasting error and detections with low forecasting error are often also interesting.

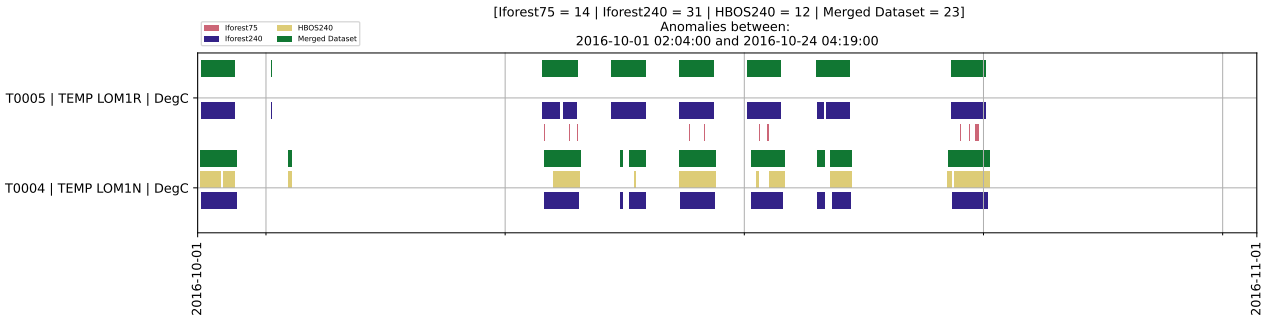


Figure 5.19: A comparison of detections made using three different detectors and a resulting merged dataset. Nearby anomalies as well as the detections accross different scorers are merged into a singular detection.

The locality attribute, used in ESA-ADB to denote whether the values of an anomalous event are within the nominal values of an anomaly, is not represented in the metrics for this project. The nominal values for each channel may vary over time and true nominal values are not provided with the data.

The final set of values stored for each detection is shown in Figure 5.20. If eventually used to publish an anomaly benchmark, the anomalies found should be manually vetted in collaboration with instrument engineers. Additionally, start and end times precisely adjusted to match the actual anomalies, avoiding the common ”F3: *Mislabelled Ground Truth*” flaw discussed in Section 2.2.3. Finally, known rare nominal events and anomalies should be added back to a complete anomaly table and should be distinguished using an additional column such as ”class” and ”subclass” used by ESA-ADB. In mask format, rare nominal events could be stored in separate masks to facilitate better model performance metrics.

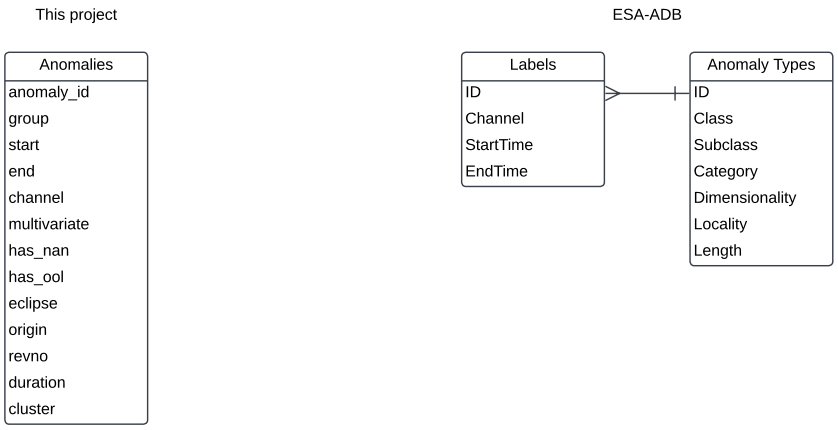


Figure 5.20: Full format of anomalies in tabular format for this project compared to ESA-ADB.

5.5. SHAP analysis

As described in Section 2.3, SHAP can be used to interpret which covariate channels contribute most to anomalous behaviour. Time constraints meant that the developed SHAP methodology could not be fully realised or used to analyse detected anomaly candidates. Instead, this section documents the methods explored as well as an incomplete implementation to facilitate future work.

The usefulness of a SHAP explanation is dependant on how anomalies are encoded into the machine learning model. Three implementations were considered throughout the thesis:

S1. **Direct application on the forecaster** — In initial experimentation, SHAP was applied to the forecasting model directly. Interpretation is focussed on describing the contribution of each covariate to a certain forecasted telemetry value. The SHAP values of an anomalous range of telemetry data may then be compared to the nominal behaviour to find a potential cause. The problem with this method is that an ideal semi-supervised model always predicts nominal behaviour, resulting in an approach that is unsuitable for interpreting anomalies. Nevertheless, this implementation can still be used to get an indication of which covariates drive forecaster behaviour the most.

S2. **Indirect application 1**: Second model for **predicting** anomaly scores — This method is developed by Gomez et al. (2024) [12] and was shown to be useful for finding which covariates influence temperature anomalies in the Euclid space telescope. An anomaly score is retrieved from the forecasting error and an XGBoost model is trained to predict the anomaly score. As input, covariates are provided at three time lags (30 minutes, 4 hours and 24 hours). The target channel itself is not provided to force the model to learn relationships between the anomaly score and other channels. Assuming an accurate anomaly score, SHAP values then directly describe a covariate’s contribution to the anomaly. Unfortunately, the use of XGBoost autoregression means the method does not scale well to larger datasets (as found in Section 5.2.3). To account for the time constraint in the thesis some modifications were made to improve computation times.

S3. **Indirect application 2**: Second model for **classifying** anomaly detections — This is the proposed evolution of the previous method. The previous implementation is converted from an autoregression to a supervised classification problem to better utilise the detections made using the pipeline. Covariates are supplied to an XGBoost classifier, which is trained to mark time steps as anomalous or not anomalous.

The use of XGBoost is maintained to utilise the fast *TreeExplainer*, discussed in Section 2.3. To account for an imbalanced dataset, anomalous segments are provided with a sample weight that is equal to the ratio of anomalous vs nominal instances, encouraging the model to focus on anomalous sequences equally. Using XGBoost directly instead of relying on the Darts autoregression system allows the total computation time to be reduced to minutes instead of an hour. Additionally, the simplified encoding allows for straightforward aggregation: SHAP values can be filtered for areas where a pipeline detection is present.

The three SHAP encodings and their connection to the intermediate pipeline outputs are shown in Figure 5.21

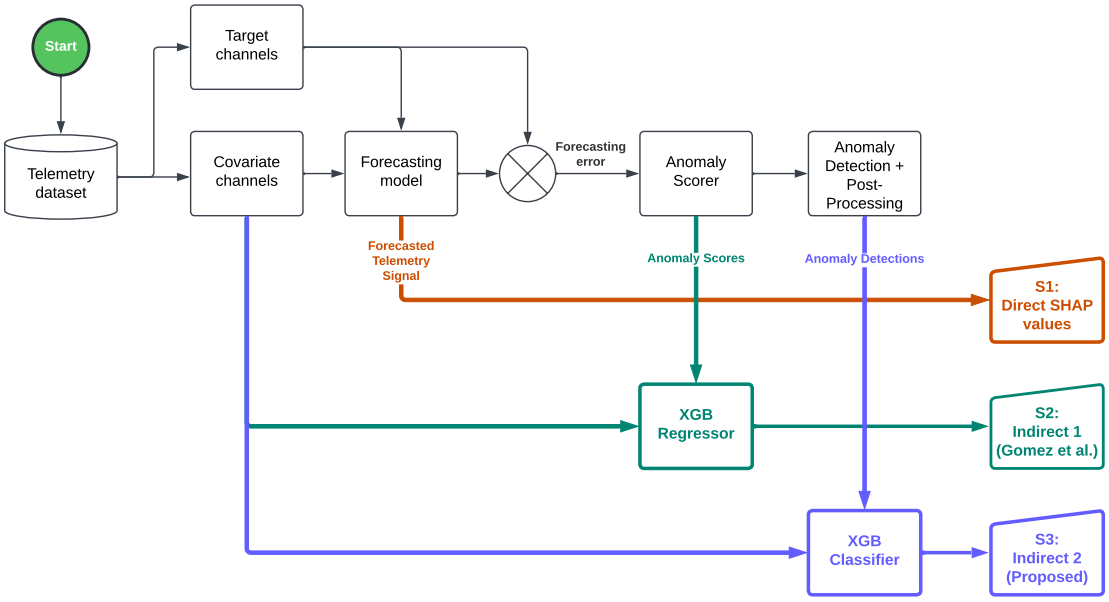


Figure 5.21: Connections between the forecasting model derivatives and the various SHAP models explored.

A singular attempt to apply method **S3** on the PN CCD temperature (F1129) yielded limited success. A number of PN channels of various units were selected, avoiding those that are localised to PN's quadrants. The time lags used were 15 minutes and 60 minutes. As before with the forecasters, the XGBoost classifier requires a training and test set. To allow complete coverage of the detections but still avoid data leaks, the dataset was split and trained on two models, following the scheme of Figure 5.22. Unfortunately, this meant that the models have differing classification performances as well as differing SHAP values. Methods to correct for these issues were not investigated during this thesis and should be considered in future work.

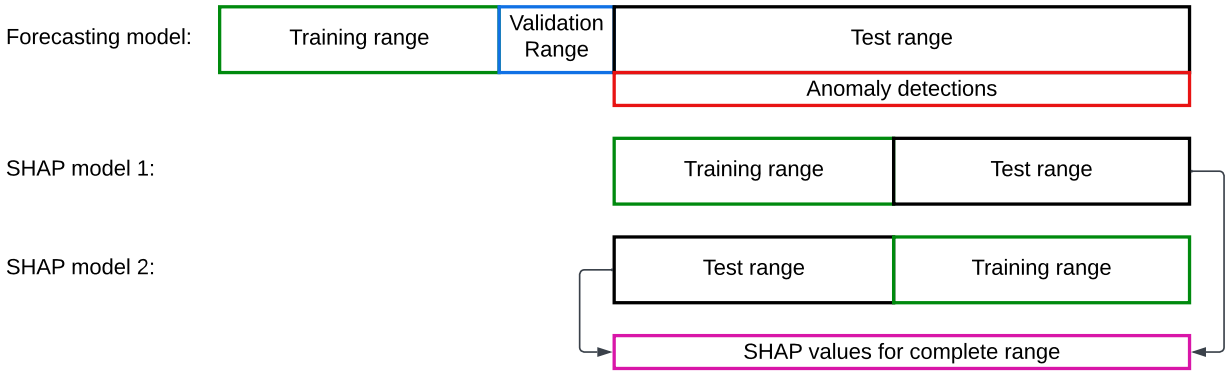


Figure 5.22: Training scheme to retrieve SHAP values for the complete range of detections using two models.

The resulting SHAP values can be visualised in various ways to interpret anomalies. Aggregated box plots, like the example shown in Figure 5.23, indicate which covariates contribute most across all classified anomalies. In this case covariates are aggregated per channel across all time for all points where the data is marked anomalous. Heatmap plots, such as Figure 5.24, show the contribution of each covariate in a localised manner.

The resulting plots are easy to understand: channels with extreme positive SHAP values contribute the most to an anomalous classification, while those with values close to zero contribute the least.

Interpretations can only be trusted if the underlying model is accurate. The ROC AUC can be used to measure the performance of the two models. Values range between 1 for a perfect classification performance, to 0.5 if the performance is as good as random, and finally below 0.5 if worse than random. Brief experimentation yielded ROC AUC scores between 0.5 and 0.8, with a large number of pipeline detections classified incorrectly. Thus, while run times are improved, some work is required to make the method usable. Unfortunately, the incomplete state of the method makes it difficult to provide concrete suggestions for improvement. Initial efforts could focus on a more careful selection of covariates and time lags as well as refining anomaly detection start and end times to improve the supervised labels. Ultimately, a return to the anomaly score regression method (**S2**) may yield improved results with less effort.

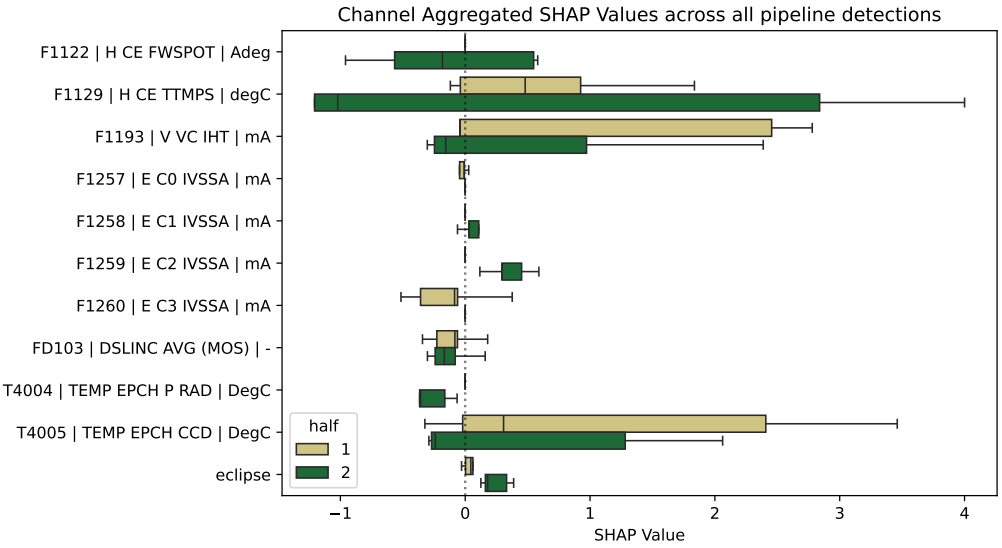


Figure 5.23: SHAP values aggregated by channel across all time lags. Instances with nominal values are filtered out, allowing for easy determination of which channels contribute most to an anomalous classification. Aggregation is performed per half.

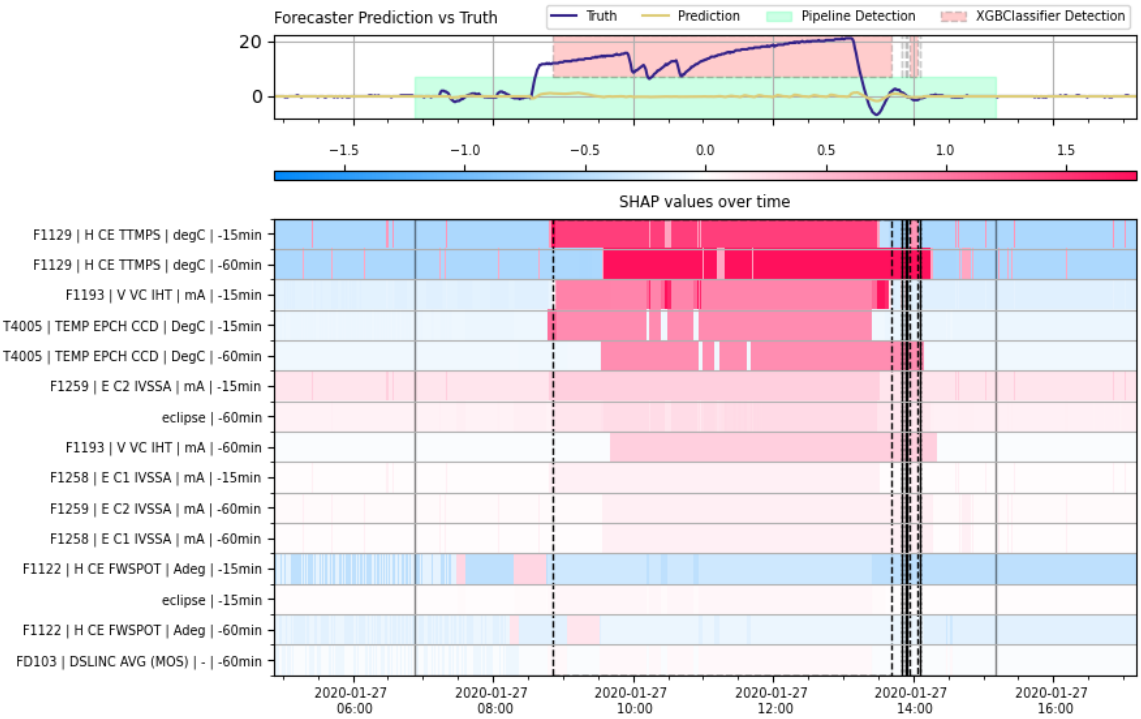


Figure 5.24: Heat map showing covariate SHAP values over time for an anomaly detected with the pipeline. The pipeline and XGBoost classifier detections are highlighted.

6. Results

In this chapter, the anomaly detection output of the thesis project is presented and analysed. Comparisons with existing anomalies allowed for refinement of the scoring system, facilitating its completion and integration into the pipeline. Because it was only completed late into the thesis, relatively few channels were analysed. Section 6.1 provides an overview of the detections. Recurring events and likely forecasting artifacts are filtered out allowing for a high level analysis of all remaining detections.Section 6.2 compares the findings to the anomaly reports presented earlier in Section 3.2 and takes a closer look at a small selection of anomalies.

A complete understanding of a detection requires a significant amount of additional information, such as a view of the active modes, the anomaly score and relevant covariates. This is provided in the form of anomaly plots in Appendix A to keep this chapter concise. Additionally, the multiple months spent looking at XMM telemetry data still pale in comparison to the rigorous training and years of experience which satellite operators and instrument engineers use to analyse anomalies. As such, any interpretations provided in this chapter should be considered with limited credibility.

6.1. A global overview of pipeline detections

The anomaly pipeline was run on a number temperature channels of OM and PN, shown in Table 6.1, with varying outcomes. The following observations can be made about the results:

- **T0004 and T0005** — These are the first channels that yielded some success on the pipeline. The discovery that local OM channels (e.g. H51XX) may occasionally become unavailable while the spacecraft has an issue (see Section 3.1.5), led to a shift in focus to thermal subsystem channels. The channels are stationary over many years and show nearly identical behaviour. As such they will be treated as jointly in subsequent analysis.
- **OM thermistors (H51XX)** — All these channels have a cyclical pattern, similar to T0004 & T0005. For months, these were the focus of the project with no success. The discovery of the previously mentioned unavailability property ultimately led to focus being shifted elsewhere. Reapplying the finalised pipeline found very few events with high forecasting error, leading to hundreds of small forecasting error-related detections which were difficult to sift through.
- **T2009, T2013, T2017, T2028** — Similar to T0004 & T0005, but containing features that require a wider context window. Only a brief attempt was made at running these through the pipeline and a second try (especially on T2013 and T2028) could yield better results.
- **T4004, T4005** — These channels only change over long periods of time and may require an alternative preprocessing approach such as discussed in Section 5.2.3.
- **F1128 and F1129** — A second set of anomalies was found in these channels very late in the thesis. Both are related to the PN CCD and are also very stationary. The results of F1129 are not included because of time constraints.
- **PN Quadrant temperatures (F1X76)** — These channels are related to the CCD quadrants of PN and are highly discretised, which may be the reason why they are difficult to forecast. Many early attempts, using various preprocessing steps, focussed on these channels with limited result. Once anomaly reports were provided, these were abandoned in favour of OM related channels.

A table with settings and models used for channels with successful detections can be found in Appendix A. The inability to forecast certain channels with the techniques applied in this thesis should not be interpreted as definite proof against their forecastability. In fact, due to the late arrival of initial results meant some of the listed channels only received a cursory attempt at anomaly detection. Ideally, these channels should be revisited once the method is improved further.

Table 6.1: Channels analysed and whether the pipeline applied was able to forecast the telemetry and find anomalies or not. Anomalies were found in F1129, but there was no time to analyse the results.

Instrument: OM			Instrument: PN		
Channel	Forecastable?	Anomalies?	Channel	Forecastable?	Anomalies?
T2028	no	-	T4005	no	-
T2017	no	-	T4004	no	-
T2013	no	-	F1129	yes	yes
T2009	no	-	F1128	yes	yes
T0005	yes	yes	F1576	no	-
T0004	yes	yes	F1676	no	-
H5135	yes	no	F1776	no	-
H5130	yes	no	F1876	no	-
H5125	yes	no			
H5120	yes	no			
H5115	yes	no			
H5110	yes	no			
H5105	yes	no			

The overview in Figure 6.1 shows all detections made. For F1128, two large concentrations are present in the middle of 2016 and 2017. Most of these detections (57/74) were found to be of the same variety as the examples shown in Figure 6.2, appearing right after a communication loss after a periapse pass. These 'periapse spikes' could be related to incomplete ground coverage during this period (discussed in Section 2.1.4 and Section 3.1.5) and amended later, although this has not been confirmed with the instrument engineer.

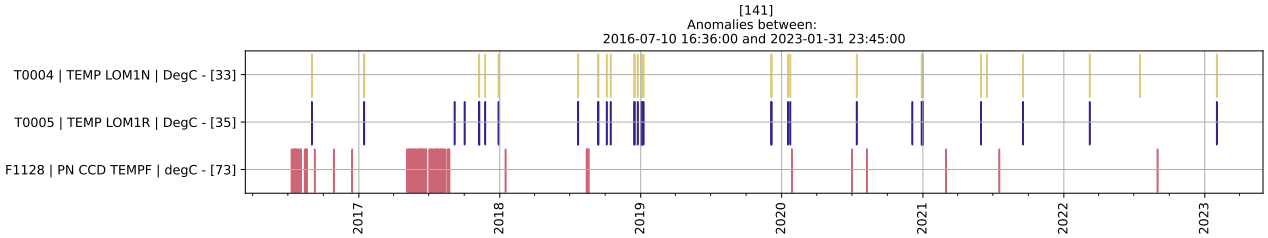


Figure 6.1: Barcode plot showing the unfiltered set of detections.

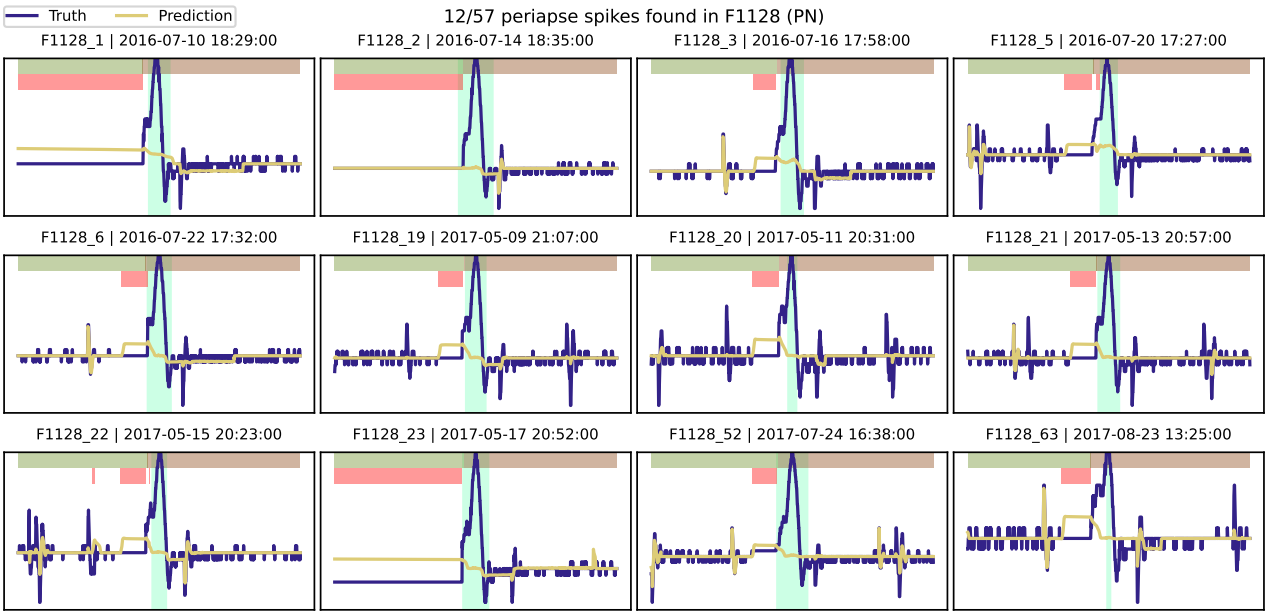


Figure 6.2: Various examples of periapse spike detections found for F1128. The alternating green and red bars at the top indicate different orbits, red bars indicate missing data.

Not every detection is an anomaly: T0004 and T0005 contain a number of forecasting artifacts where the

autoregression becomes noisy, Figure 6.3 shows some examples. At those events, the measured data does not show any anomalous change in behaviour. The removal of these artifacts along with the previously mentioned periapse peaks, yields a much more manageable number of candidate anomalies shown in Figure 6.4. The resulting anomaly candidates for T0004 and F1128 are shown in Figure 6.5 and Figure 6.6 respectively.

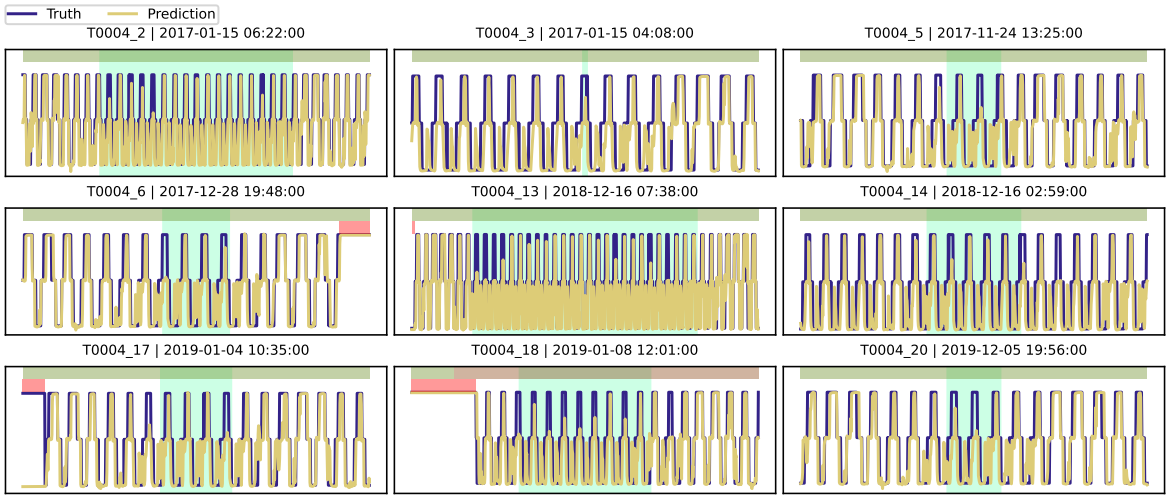


Figure 6.3: Various examples of forecasting artefacts found among the detections of T0004. The ed bars at the top indicate missing data, alternating green and brown bars indicate different orbits.

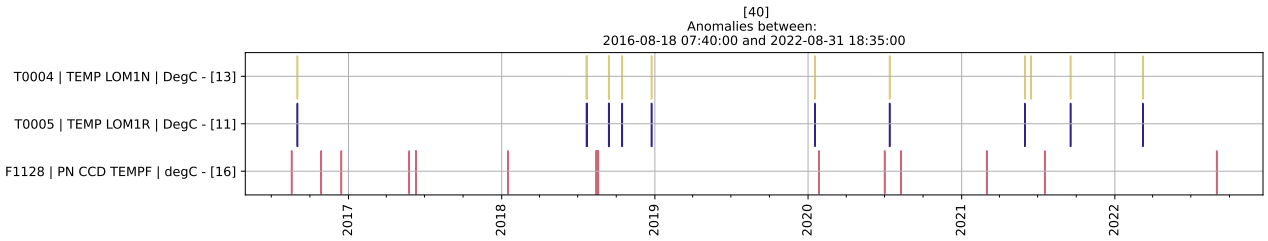


Figure 6.4: Barcode plot showing the filtered set of detections, without forecasting artefacts and the periapse peaks.

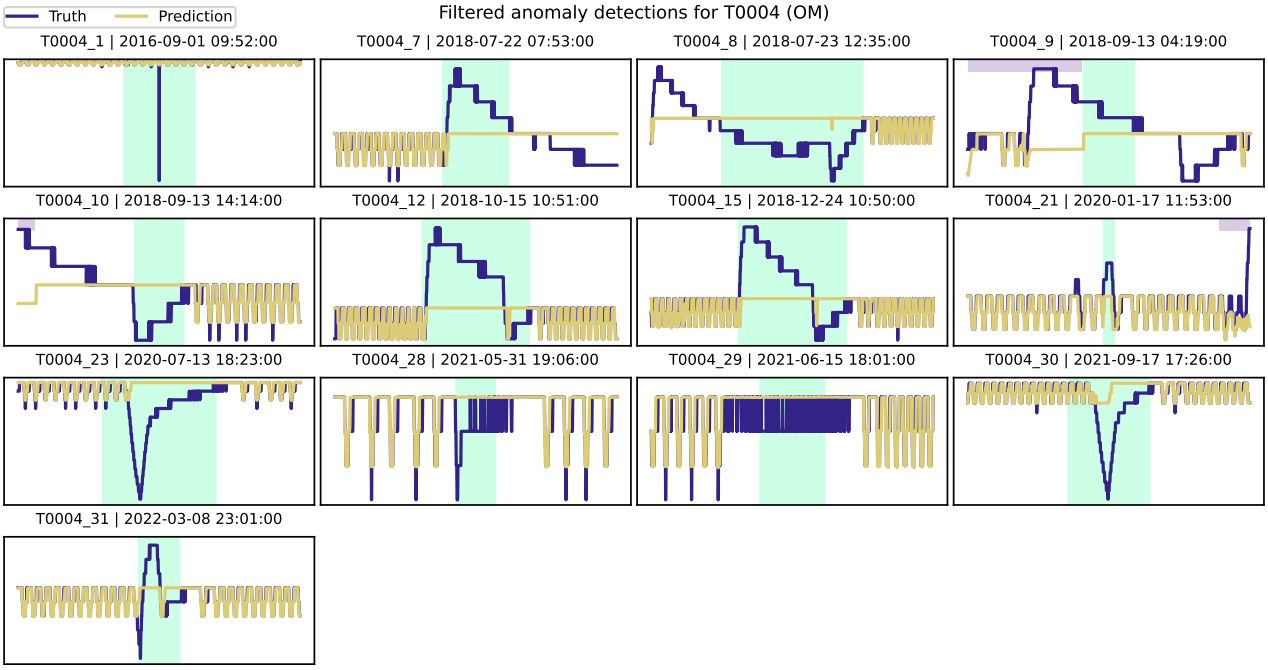


Figure 6.5: Simplified overview of the filtered set of detections for channel T0004. The purple bars at the top indicate an eclipse.

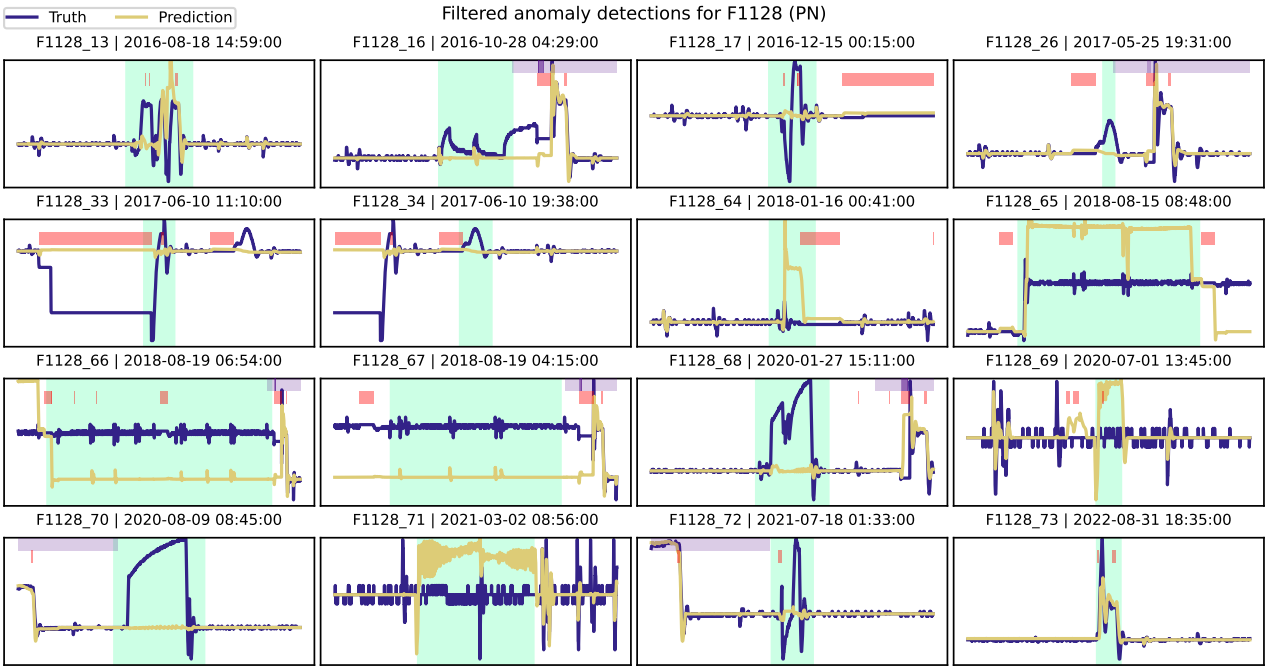


Figure 6.6: Simplified overview of the filtered set of detections for channel F1128. The purple bars at the top indicate an eclipse, red bars indicate missing data.

Even when reduced, the number is too extensive to cover each one individually and instead, some general observations are made:

O1. **Detections over time** — In Figure 6.4, the number of detections over time does not appear increase or decrease over time.

O2. **Unmerged anomalies** — A number of large nearby detections (e.g T0004_9/10 in Figure 6.5 and F1128_65/66/67 in Figure 6.6) are not combined. In fact, F1128_66 and 67 overlap each other. A clearer look at T0004_9 is shown in Figure A.2.

O3. **Similar behaviour of T0004 and T0005** — All but one detection (T0004_29 in Figure 6.5) in T0004 is shared with T0005, reflecting their similar behaviour. A closer look reveals that this 'unshared' detection is also present in the second channel, but that its flatter signal makes it more difficult to detect.

O4. **Measures against eclipses may cause anomalies to be missed** — The area before right before T0004_9 (seen in Figure 6.5) is not included the detection because the forecasting error presented to the scorer is set to zero in an eclipse.

O5. **Hallucinations caused by covariates** — F1128 (detections shown in Figure 6.6) is run with a number of additional covariate channels. Detections F1128_69 and F1128_71 occur during the rare appearance of the 'UNKNOWN' instrument mode and the unlearned behaviour causes the forecaster to behave erratically. F1128_64 (see Figure A.3) and F1128_73 (see Figure A.6) show a pattern similar to the eclipse procedure (seen occurring after the detection of F1128_26) even though no eclipse is recorded at these times.

O6. **Mostly large events** — Almost all detections have long durations and clear prominence beyond the nominal behaviour of the channel. By the taxonomy of ESA-ADB discussed in Section 2.2.4, these can be described as global anomalies. Some notable exceptions are T0004_28 and T0004_29 (all shown in Figure 6.5). T0004_1, and T0004_21 feature very short events but still go beyond nominal temperatures.

For **O1**, concluding that the number of *anomalies* over time remains constant would be incorrect as not all detections are confirmed as actual anomalies. Furthermore, the limited number of channels analysed prevents a broader conclusion on the health of the instruments over time.

For **O2**, a closer look revealed a bug in the merging step introduced in Section 5.4, causing some cases to be handled incorrectly. The threshold to combine detections could also be increased. Observation **O3** is caused by deficiencies with the scoring system, which is discussed extensively in Section 5.3.2. Similarly, **O4** can be seen as an unfortunate side-effect of the current scoring system, which sets the anomaly score to zero for eclipses and missing data. To keep manual analysis time low, eclipses must be filtered out in some way. The old technique that filtered out detections that are coincident with eclipses would have removed the detection

completely. Ultimately, neither solution is ideal, and an optimal solution would require the forecaster to correctly predict the eclipse sequence for each channel.

The first part of **O5**, regarding the appearance of the rare UNKNOWN instrument mode (see Figure A.4 and Figure A.5) and can be mended by removing appearances of this mode from the training data using sample weights. The second part is more interesting. To start, even though the model is trained with sample weights on, it is still able to predict the behaviour of an eclipse signal quite well. It may be that one or more eclipse procedure tests are present in the training data and that the model is able memorise the behaviour through the covariates, with F1128_73 then being another such test. Indeed, a closer look found that the behaviour is mirrored in F1129 for F1128_73 (see Figure A.6) while F1128_64 (see Figure A.3) contains a voltage spike in F1198 which also occurs during eclipse events. This sparks an interesting point about the use of covariates, which will be discussed further in Section 7.1.4.

Finally and most importantly is observation **O6**. As the anomaly pipeline presented in this thesis and all other machine learning-based anomaly detection algorithms effectively compete with existing systems like out-of-limits, it would have been more interesting to find a larger number of inconspicuous anomalies. While this is not entirely unexpected given the discussion on the limitations of the scorer in Section 5.3.2, the type of anomalies found are also partly attributable to the channels analysed. For F1128, no other anomaly type is possible as its nominal behaviour is essentially a constant temperature. T0004 and T0005 played a prominent part in this thesis (see Section 5.3.1) and manual analysis of their forecasts yielded no other significant events similar to T0004_28 and T0004_29. The high amount of discretisation in these channels means there is very little opportunity for anomalies to occur within nominal bounds. In other words, while this type of anomaly is definitely a weakness of the pipeline, they are also simply less common in the channels analysed.

In Section 6.2, a closer look is taken at F1128_16, F1128_65/66/67, T0004_12 and T0004_29.

6.2. A closer look at pipeline detections

In this section, a selection of anomalies are considered individually. In Section 6.2.1, the output of the pipeline is compared to reported anomalies from ARTS. Section 6.2.2 presents the anomalies discussed with instrument engineers. Finally, some additional recurring anomalies are discussed in Section 6.2.3 .

6.2.1. Comparison with anomaly reports

Figure 6.7 compares the detections from the previous section to the anomaly reports, showing that only three events (plus one duplicate) overlap. As described in Section 3.2 however, only twelve reports show behaviour in the channels analysed, with only seven (plus the duplicate) in the test set range between 2016 and 2023. Of those events, shown in Figure 6.8 only three resulted in an actual detection.

A more detailed view of the anomalies is required to provide real interpretations on the missed detections and a curious reader may consult Appendix A.3 for a closer look. Although IOPS-35 and IOPS-36 are directly linked to PN, the effect of the anomaly is much more pronounced in other channels. Similarly, report SC-108 is related to an event in the AOCS system and the spike and subsequent missing data may be indirect consequences in the rest of the system. These three anomalies are not caught for two reasons: the forecaster predicts the spikes because they are also present in the covariate channels and the anomaly score is not prominent enough to be detected. It can be debated whether, SC-108 and IOPS-36 as manifested in F1128 should be considered anomalous at all. Nearby spikes of similar appearance did not require a report for example. Nevertheless, it again sparks the question of whether covariates provide too much information to the forecasting process, which is discussed further in Section 7.1.4. Finally, SC-145 again shows the property where a channel returns missing data if there is a problem deeper in the satellite. The scoring and detection thresholding could again be improved to yield a better result here.

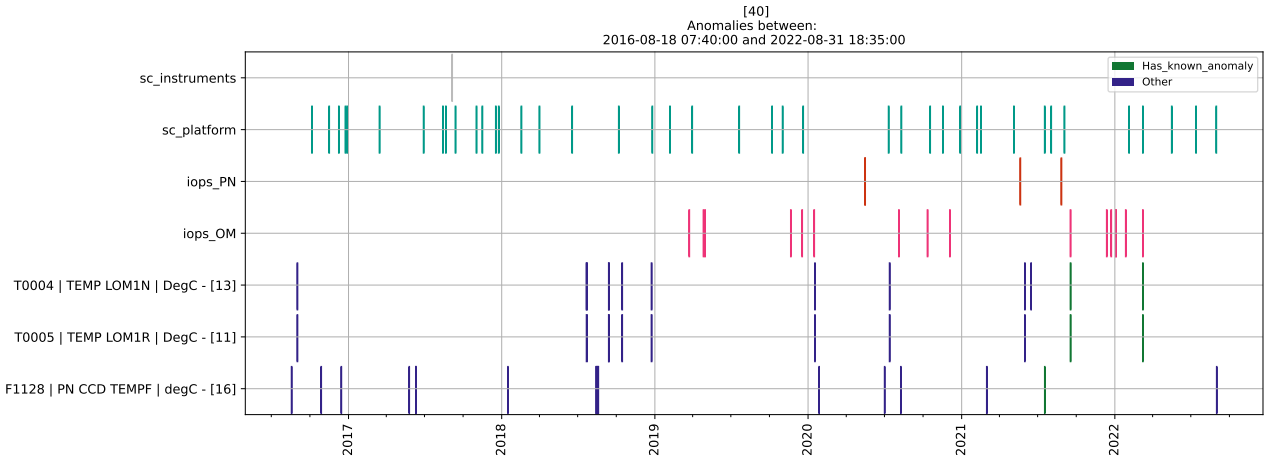


Figure 6.7: Barcode plot comparing detections of the pipeline to the anomaly reports. For the three analysed channels, legend colours denote whether detections have a corresponding anomaly report.

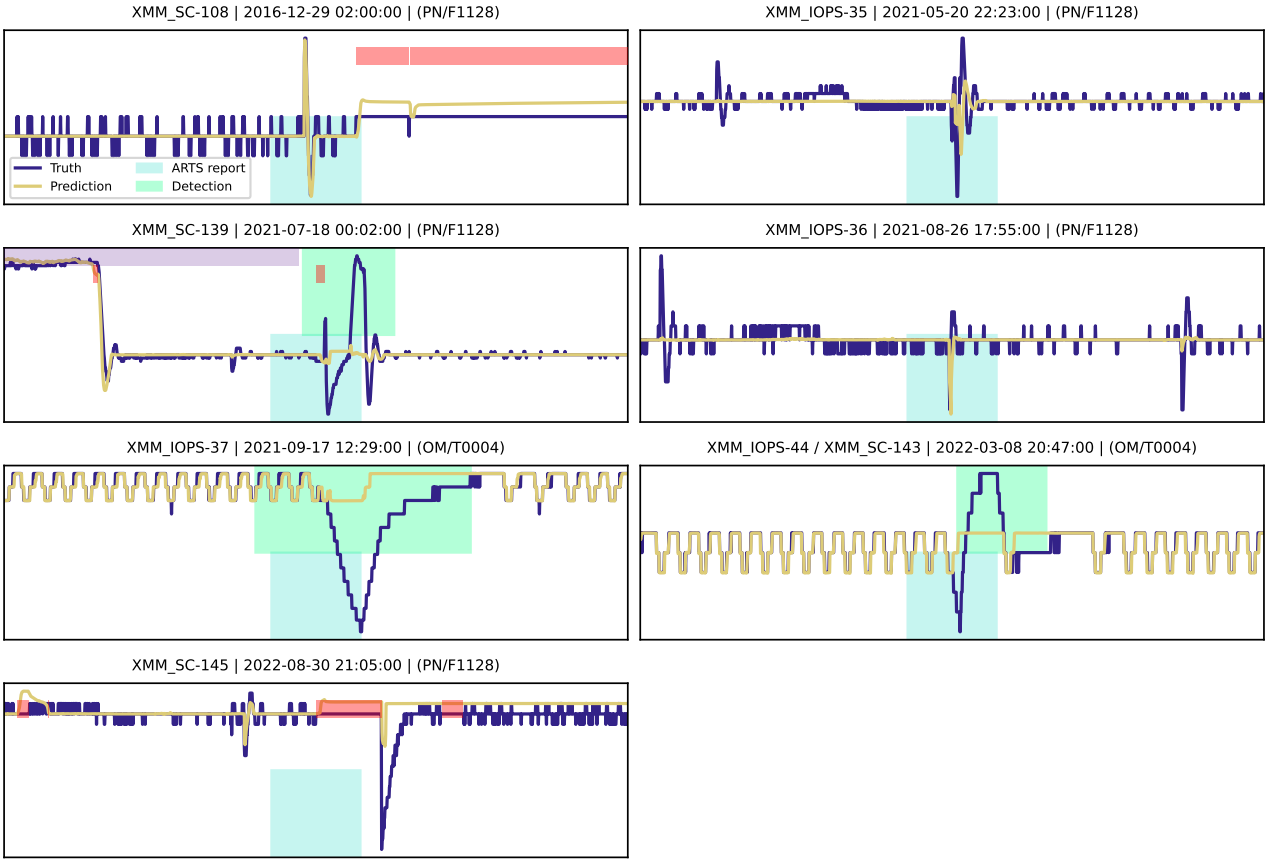


Figure 6.8: Comparison between reported anomalies and their detection by the pipeline. The centre of the blue bar marks the reported start time of an anomaly event while the green marks a detection. Same as previously: red and purple bars indicate missing data and eclipses.

6.2.2. Anomalies discussed with engineers

As discussed in Section 4.1.3, anomaly reports in ARTS do not contain all the anomalies encountered by spacecraft operators. To check whether the other detections could be considered as anomalies, a few of the more striking ones related to PN were discussed with the instrument engineer:

1. **F1128_16** — Is related to an unexpected activation of CCD heaters caused by a radiation event elsewhere in the system. The detection is shown in detail in Figure 6.9.
2. **F1128_65/66/67** — Is related to an incorrectly passed command set to test a new eclipse procedure. The CCD is heated to eclipse temperatures but could not be returned to ideal until the next eclipse. The forecaster is able to predict the jump due to the addition of covariates, but cannot find the right temperature as the commanded temperature and heater channels are not provided as covariates.

Although not all detections could be discussed, both of the detections presented can be seen as anomalies and serve as an indication for the validity of the results of the pipeline.

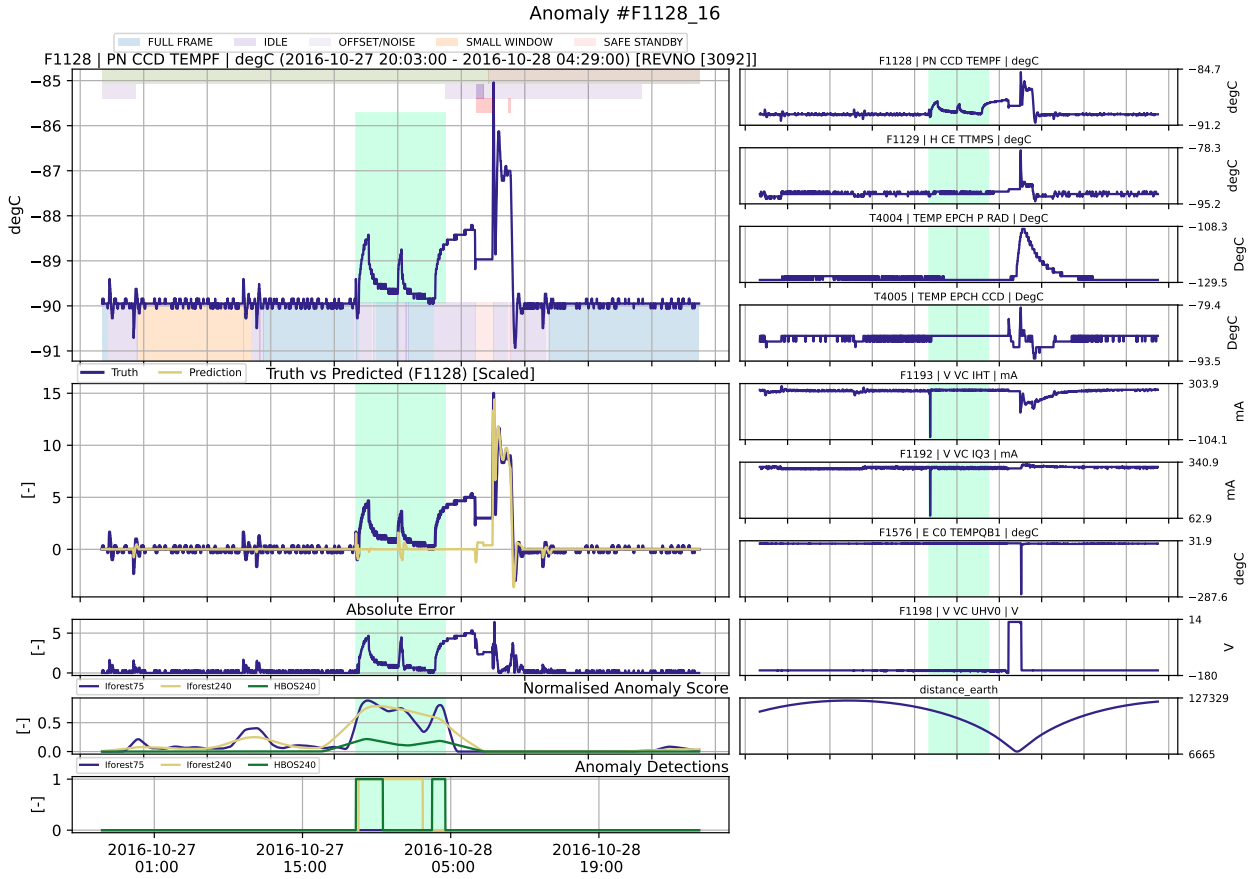


Figure 6.9: The full anomaly plot for detection F1128_16 showing information related to the scoring, detection and additional covariates.

6.2.3. Recurring anomalies

Two recurring detections are shown in Figure 6.10 and Figure 6.11. Both events coincide with a switch to safe mode and/or effects related to the power channels (P1035 and P1135). In fact, almost all detections related to OM contain either an LCL trip (P1035) or some abnormal readings in the current sensor (P1135), which leads to questions whether a check for those channels might be a more efficient way to detect problems in OM. Because widespread access to platform-related channels is currently unavailable through ARES, it is currently not really possible to see whether the event is caused by factors within the rest of the spacecraft, or if they start at OM. The addition of platform-related data might thus be beneficial in continued anomaly detection efforts.

The coincident switch to safe mode does spark the question whether these detections are actual anomalies or just planned events. Confirming such cases, requires them to be presented to the instrument engineers but in worst case, they can be marked as rare nominal events.

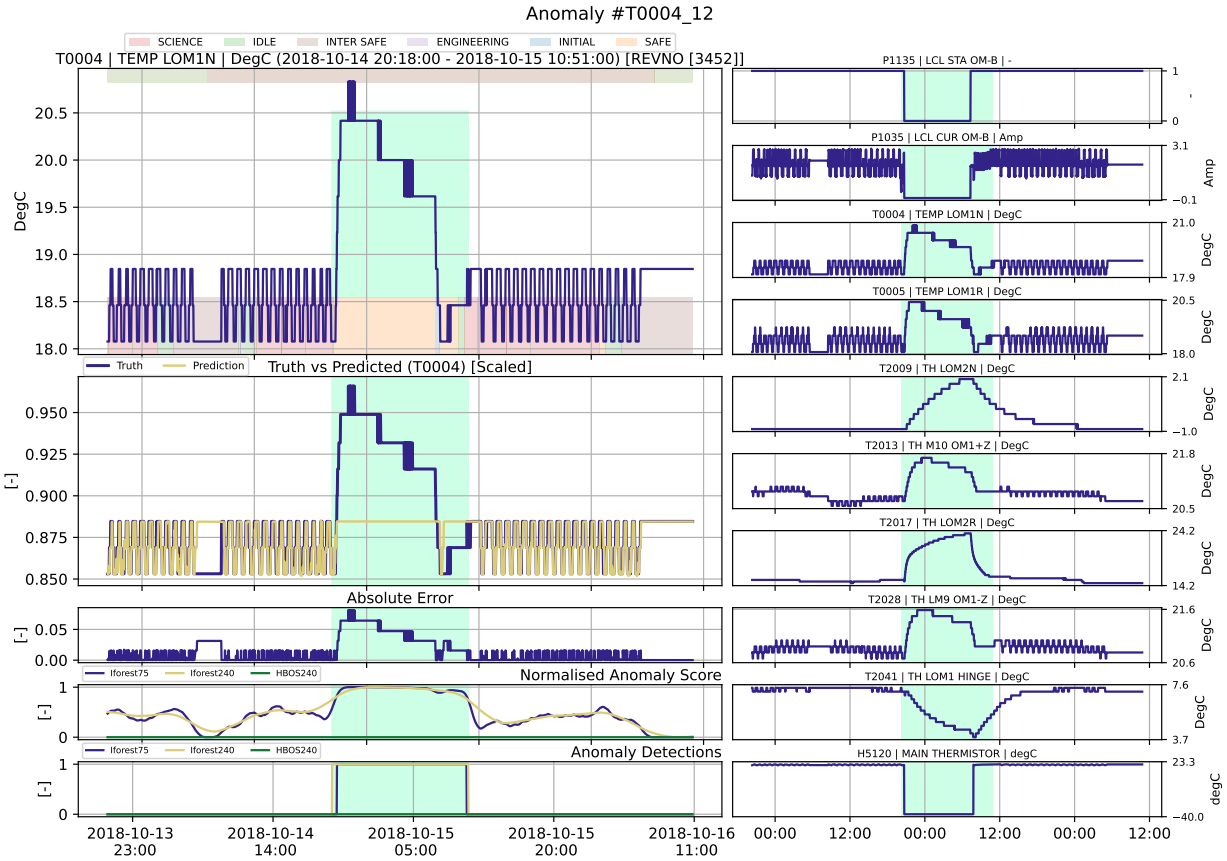


Figure 6.10: The full anomaly plot for T0004_12.

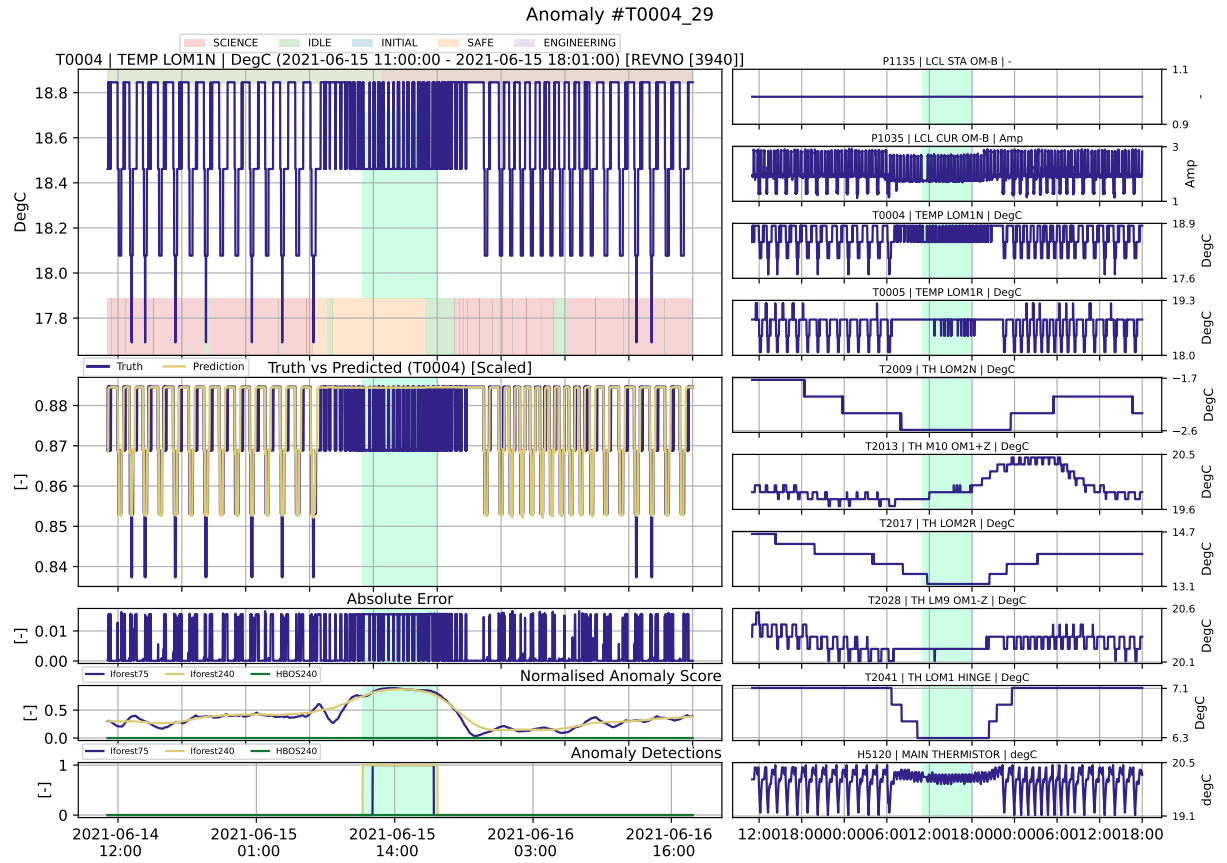


Figure 6.11: The full anomaly plot for T0004_29.

7. Discussion and Future Work

The results of the previous chapter demonstrate that the pipeline and the overall methodology are capable of detecting anomalies. However, the approach still contains a number of limitations and flaws discovered throughout the thesis. A collection of these themes are discussed in Section 7.1 and, where relevant, suggestions are presented for improvement. In Section 7.2, the initial approach is updated to streamline development towards an eventual XMM anomaly benchmark.

7.1. Areas for discussion and improvement

This section covers a number of areas for discussion and improvement, grouped into a series of themes. Where relevant, comparisons are made to the methods used by ESA-ADB [59][18]. Related to the pipeline are Sections 7.1.1 on data retrieval and preprocessing, 7.1.2 on sample weights, 7.1.3 on the overabundance of parameters, 7.1.4 on covariates, 7.1.5 on scoring and 7.1.6 on detection post-processing. Section 7.1.7 discusses the anomalies found using the pipeline. Lastly, focussing on future additions, are Sections 7.1.8 on SHAP, 7.1.9 on manual refinement and iterative improvement, and finally 7.1.10 on the use and collection of additional anomalies.

7.1.1. Data retrieval and pre-processing

As identified in Chapter 3, XMM-Newton telemetry data contains a number of challenging features, such as eclipses, a high amount of missing data and a number of non-ordinal mode channels. All of these were discovered incrementally throughout the thesis and mended through various pre-processing steps which are combined into a telemetry dataset tool presented in Section 5.1. Developed with continued use after the thesis in mind, the tool is efficient, well-documented and highly customisable. It is already being used experimentally at ESAC and is ready to be used as part of an anomaly benchmark.

A few very small improvements are suggested in Section 5.1.6. The addition of other data sources like platform-related telemetry, recent orbital data, telecommands and more could be considered in future work. Much of this data is currently not stored in ARES or at ESAC, as it is not directly relevant to instrument health / science data, and would potentially have to be obtained from the operations team at ESOC. The addition of telecommands would be especially useful, as discussed in Section 7.1.4.

7.1.2. The use of sample weights

The semi-supervised forecasting models used in this thesis nominally require anomaly-free training data and the use of sample weights were found to be a good method to achieve this. In comparison, ESA-ADB removes anomalous segments entirely. Applied to XMM, with its frequent missing data points and eclipse events, would result in a highly fragmented training set. Very small segments of training data would have to be discarded, as they could not provide sufficient lookback data. While the sample weight method allows for the models to be supplied with a single continuous segment of training data, it could also introduce compatibility issues. Common deep learning frameworks such as Tensorflow and PyTorch (which Darts uses) as well as a number of other machine learning libraries like XGBoost support sample weights natively. For semi-supervised algorithms that do not require lookback, anomalous data could simply be removed as done in ESA-ADB. Other algorithms without sample weight support would require another solution to be made compatible.

Currently, timestamps containing eclipses and missing data are weighted zero, but this should be expanded to include known and newly detected anomalies as well. The iterative addition of newly found anomalies is a method which was already used to create ESA-ADB and could serve to significantly improve forecasting performance in a continuation of this project. To facilitate this, anomaly detection should also be performed on the span of data currently used for training. In the future, the forecasting error for the full dataset could be retrieved by splitting the full set in two, and training and forecasting using two separate models. A similar scheme was already presented in Figure 5.22 for the implementation of SHAP.

7.1.3. The overabundance of channels, variables and hyperparameters

Applying forecasting models yielded usable reproductions on a number of channels. Nevertheless, the abundance in choice in targets, covariates, preprocessing settings, models and their hyperparameters made the process a time-consuming endeavour. The hyperparameters can be fixed by applying model tuning but the others are a bit more difficult. To start, the possibility to configure the dataset tool, most notably through the collation period, provides additional methods of improving a forecast, as shown in Section 5.2.3. However, the

lack of a single best preprocessing recipe for all channels creates a significant amount of additional work to set up a forecast. Additionally, due to varying levels of availability and discretisation, not all channels are easy to forecast and arguably, not all should/need to be analysed. As an example, PN contains a number of channels that collect individualised information on each of its four CCD quadrants. If the channel that measures the 'parent' power line shows an anomaly then the issue will likely be visible in the quadrant lines that it feeds as well. The parent line should then be preferred for analysis if it is easier to forecast, especially since the more localised telemetry channels often lack data when there is a problem elsewhere in the satellite (see Section 3.1.5). A similar case was found in Section 6.2.3, where a number of OM-related detections were found to be related to power fluctuations and LCL trips.

To streamline the process in the future, channels should be selected based on discussion with instrument engineers and/or spacecraft operators, focussing on their importance to the instrument and likelihood to display anomalies. Additional instruments as well as platform related telemetry connected to the instruments could also be considered. These channels should then be tested on their ability to be forecasted using a standard pipeline setup. Those that can't, should be discarded until a desirable number of benchmark target channels are collected (around 50 in the case of ESA-ADB). Specific setups can then optionally be created for batches of target channels to improve individual detection performance. A set of existing anomalies for each case could then be used to tune hyperparameters and other values.

7.1.4. The use of covariates

In Section 5.2.3, covariates were shown to have profound effects on forecasting capabilities, but not all channels provide desirable effects. The usefulness of modes is clear, as some patterned channels, like T0004 and T0005, can be predicted using the primary instrument mode and the heater status alone. The use of other channels had mixed effects. Initial experimentation used all non-target channels of an instrument as covariates. The assumption being that the forecaster would not be able to learn complex and seasonal behaviours without additional context. The discovery that an overabundance of covariates produced additional noise and thus worsened forecasting resulted in a change of doctrine: only the covariates that were shown to improve forecasting performance were kept. This still resulted in a different problem shown in Chapter 6: models were able to partially predict anomalous events due to their presence in other channels.

Preventing this can be done by choosing covariates based on their physical connection to the target, limiting what is learned by the forecaster using sample weights, or where possible, avoiding the use of covariates entirely. Eventually, a replacement with telecommands, which provide context without an additional anomalous component (barring human error) would be optimal.

7.1.5. Anomaly scoring and thresholding

The anomaly scoring and detection system of the pipeline is seen as the weakest point in the thesis and its limitations are discussed extensively in Section 5.3.2. The primary issue is that the methods used are exclusively reliant on the forecasting error, which naturally reduces the chance that small, inconspicuous anomalies are caught. This is also evidenced in the detections shown in Chapter 6. Secondly, the unsupervised machine learning algorithms used for scoring are highly parametrised requiring tuning to reach optimal results. Lastly, the use of a static threshold to extract detections from the anomaly score requires additional tuning and produces detections even when only small forecasting errors are present. Thus, while applying a scoring function showed a clear benefit over a plain forecasting error, matching earlier findings [52], the implementation could use some improvement.

The use of automatic tuning could help with the selection of parameters for the scoring function and method to do this has been provided in Section 5.3.2. This would require a larger set of pre-labelled anomalies to serve as ground truth. For thresholding, the implementation of dynamic thresholding methods such as those used in the Telemanom algorithm [13], would reduce the amount of tuning required. This would have to be accompanied by some type of minimum threshold to reduce the detection of points with very small forecasting errors. Lastly, the removal of eclipses from anomaly scoring should be revisited to ensure that no detections are missed as seen in **O4** in Section 6.1.

It should be noted that the use of unsupervised TSAD algorithms to retrieve an anomaly score from the forecasting error is a fairly uncommon practice. In fact, many existing TSAD algorithms simply use the plain forecasting error or feature a custom scoring system. Even though a scoring function can be seen as a modular component that can be 'attached' to any forecasting or reconstruction model, there has been very little focus on the development of model-independent scoring functions [52]. In the long term, the development of such functions that can also account for the local context of the telemetry signal as well as its physical and statistical properties could be of great benefit to the TSAD field.

Improvements to the scoring system would be the fastest way to increase the performance of the pipeline.

7.1.6. Detection post-processing and cataloguing

The detection post-processing tools introduced in Section 5.4 create metrics for each detection and allow for the merger, grouping, and storage of anomalies. The system is inspired by ESA-ADB and should be nearly ready for the use in an anomaly benchmark.

A software bug that results in a failed merger of overlapping detections was identified in Section 6.1 and should be fixed before future use. Additionally, the inclusion of eclipses and sections of missing data in a future benchmark will require an additional metric (e.g. 'type' or 'class') to distinguish them. Finally, a tool should be created to convert between mask and tabular anomaly formats.

7.1.7. The quantity and type of anomalies found with the pipeline

The results in Section 6.1 prove the ability of the pipeline to detect anomalies in XMM telemetry data. Nevertheless, time constraints meant that the pipeline was only run successfully on a few target channels. Almost all of the resulting detections extended prominently beyond the nominal behaviour of the channels. Although this can partly be attributed to the types of channels analysed, which all featured little opportunity for inconspicuous anomalies to occur, many of the detections could also be caught using simple out of limits systems and would not make for an interesting anomaly detection benchmark (per flaw **F1**: Triviality, discussed in Section 2.2.3).

Unfortunately, additional non-trivial anomalies can only be found by spending more time on improving the pipeline and running it on more target channels. Further feedback and suggestions by instrument engineers and/or satellite operators could be used to find better target channels.

7.1.8. The use of explainable AI

Multiple implementations of the explainable AI method SHAP were explored during the thesis. Unfortunately, time constraints meant a final implementation could not be fully realised. Nevertheless, it is believed that a more rigorous attempt at applying SHAP to interpreting anomalies could yield beneficial results and the methods explored are documented in Section 5.5 to facilitate future work.

7.1.9. Manual refinement and iteration

Compared to the work performed performed in this thesis, the creation of ESA-ADB involved much more iteration to produce its dataset of anomalies. Authors combined an initial set of anomalies from ARTS and other sources with a second set obtained by a first detection pass using unsupervised TSAD algorithms. The anomalies in this collection were then refined manually in collaboration with spacecraft operators to assign the correct start and end dates, as well as the correct target channels. This initial collection allowed for the generation of anomaly-free training data, which was fed to the semi-supervised Telemanom algorithm [13] to find additional anomalies. This final step using semi-supervised models was repeated a number of times with refinements and discussion with operators at each step until the training data was anomaly-free and no more detections could be found.

The extensive amount of iteration and discussion with spacecraft operators / instrument engineers was not possible in this thesis, but should be pursued when proceeding with an XMM benchmark. While a number of anomalies can be confirmed by accessing additional anomaly sources, only a the experts can confirm whether a previously undocumented anomaly is real or not. Lastly, manual refinement is required to ensure that a benchmark of telemetry anomalies meets benchmark flaw **F3**: mislabelled ground truth, discussed in Section 2.2.3. Manual refinement could also be done using a purpose-built annotation tool such as *OXI* [84].

7.1.10. The use and collection of existing anomalies

The ARTS reports provided a key source of ground truth used to tune the anomaly scoring system shown in Section 5.3 and enabled comparison of pipeline output. As mentioned previously, the existing anomalies should be removed from training data and it is beneficial to collect as many existing anomalies as possible before starting with semi-supervised anomaly detection. This can be done in a number of ways:

1. Existing anomalies stored outside ARTS could be collected and refined.
2. As performed by ESA-ADB, an initial unsupervised pass (e.g. using HBOS, isolation forest, etc.) can be used to retrieve an additional set of anomalies.
3. While direct detection methods used in this thesis are most popular and allow for a more precise annotation of anomalies, a window-based detection method could also be used. Ruszczak et al. (2023) [57] present such an approach to create an anomaly dataset in OPS-SAT telemetry data. A very simple approach could window the dataset by orbit and collect various statistical features for each. An unsupervised clustering algorithm like K-means could then be applied to find the most anomalous orbits. Such a method would still require significant additional manual refinement to yield usable anomalies.

7.2. Proceeding towards an anomaly benchmark

The primary research question of the thesis (**RQ1**) focussed on finding an approach to construct a dataset of anomalies in XMM instrument telemetry data. Section 4.2 presented the initial version implemented during the thesis (visualised in Figure 4.2). The lessons learned, which are documented in the previous section, can be used to update the approach in order to proceed towards the creation of an XMM anomaly detection benchmark. This new version is shown in Figure 7.1 and features a number of new additions:

- **Choosing targets and covariates** — As discussed in Section 7.1.3 and Section 7.1.4, discussion with engineers based on the lessons learned in this thesis can help streamline the selection of target channels and their covariates.
- **Finding and refining existing anomalies** — An extended set of existing anomalies can be used to refine the scoring system and improve semi-supervised forecasting performance (see Section 7.1.2 and Section 7.1.5). Three methods to perform this step have been provided in Section 7.1.10.
- **Manual refinement and Iteration** — Manual refinement is necessary for the detections to have correctly labelled start and end times as discussed in Section 7.1.9. Some iteration was already included in the original chart but could not be performed in this thesis. The importance of this step to improve pipeline performance is discussed in Section 7.1.9 as well.

The addition of these steps and other improvements discussed in Section 7.1 should serve to streamline the creation of a benchmark. Nevertheless, it is necessary to acknowledge that this time-consuming endeavour will undoubtedly require further refinement in the methodology in the future.

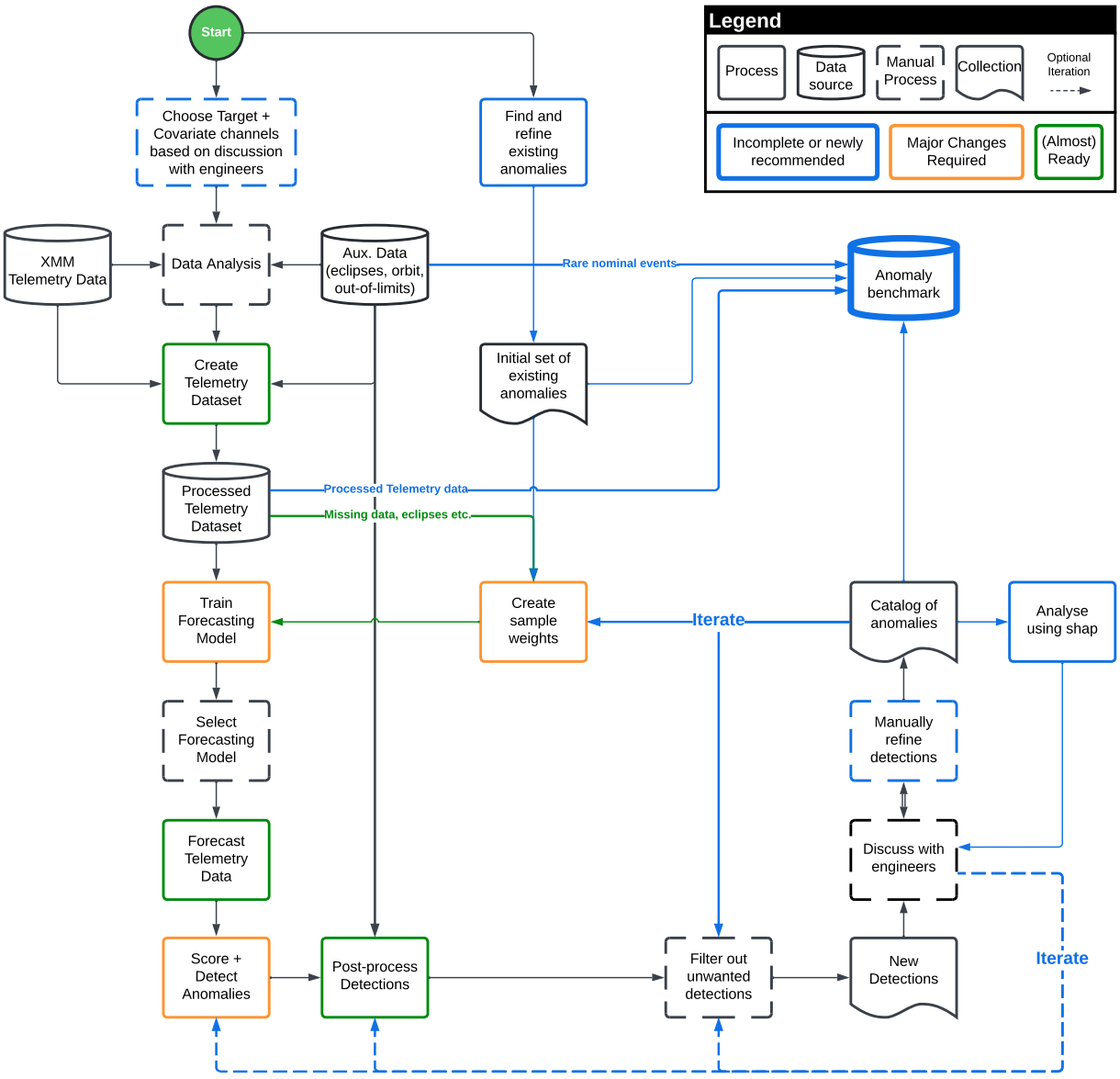


Figure 7.1: Simplified overview showing the updated methodology for the creation of an XMM telemetry dataset.

8. Conclusion

In this chapter, the conclusions of the project are presented by providing answers to the research questions and sub-questions first introduced in Chapter 1. This is done in Section 8.1. Some final remarks on the objectives of the thesis are then provided in Section 8.2.

8.1. Answers to the research questions

Research Question 1 (RQ1)

> *What is a suitable approach to construct a dataset of anomalies in XMM-Newton instrument telemetry data using ML-based TSAD techniques?*

An initial methodology to address this question was introduced in Section 4.2, focussing on a direct detection-based semi-supervised forecasting approach. The software portion of the methodology was implemented through an *Anomaly Pipeline*, presented in Chapter 5, which has four primary components:

1. Data pre-processing — Telemetry and auxiliary data is combined into a format that is digestible to machine learning models. Further explained below in **RSQ2** and at large in Section 5.1.
2. Telemetry forecasting — Pre-processed telemetry data is passed to forecasting models like XGBoost and LSTM. Multiple forecasting models were tested, finding that more than one model has good performance but that the best model depends on the channel, preprocessing and the amount of hyper-parameter tuning used. In the end, LSTM-based models were selected due to their simplicity, broad compatibility and fast runtimes. Presented at large in Section 5.2.
3. Anomaly scoring and thresholding — The resulting forecasting error is modified through various anomaly scorers. A threshold is subsequently applied to retrieve detections. The implementation is presented in Section 5.3 and features one of the major limitations of the pipeline, which is discussed further below in **RQ1S7**.
4. Detection post-processing and cataloguing — Detections are post-processed to include supplemental information. Further discussed in **RQ1S4** and at large in Section 5.4.

This approach successfully yielded a number of detections but also contained a number of flaws, which were subsequently addressed in an updated methodology presented in Section 7.2. Although it is likely that there are still improvements to be made, this approach could serve as a starting point for the creation of an anomaly benchmark in the future.

Additional conclusions are provided through the sub-questions below.

RQ1 sub-question 1 (RQ1S1)

> *What are the unique or notable characteristics of XMM instrument telemetry data and how do these affect the anomaly detection methodology?*

Based on literature presented in Section 2.2.4, spacecraft telemetry in general is known to have a number of challenging properties such as:

- A high number of telemetry channels with many interdependent features. They may contain a variety of information such as sensor data (e.g. temperatures and voltages), operating modes, status flags and counters.
- Varying sampling rates, irregular acquisition times, invalid segments, and communication gaps.
- Seasonal features related to operational modes and phases, the orbit and component degradation.

Data exploration in Chapter 3 found that all these properties are present in the telemetry data of XMM as well and are dealt with by the data pre-processing step, discussed in **RQ1S2**.

Two properties that make XMM a particularly challenging case for anomaly detection are the occurrence of seasonal eclipses (see Section 3.1.4), which require special operations, and a high volume of missing data (see Section 3.1.5). The ground operations architecture of XMM requires continuous contact with the ground for downlink and command. A lack of mass onboard storage means that a loss in signal means a loss of telemetry

data. These issues, which may be considered non-nominal telemetry behaviour, needed to be removed from training data in line with the requirements of semi-supervised anomaly detection algorithms. In the end, the best solution found utilises sample weights, which force forecasting training algorithms to ignore missing data and eclipses in the training data.

A final interesting property discovered while exploring the data is that the downlink of channels can be affected by problems elsewhere in the satellite. An example provided in Section 3.1.5 shows how an existing anomaly related to OM is measured in related thermal subsystem telemetry but contains missing data in telemetry of the instrument itself. This led to channels related to the thermal subsystem being prioritised in the search for anomalies.

RQ1 sub-question 2 (RQ1S2)

> *Which preprocessing steps are required to transform unprocessed XMM Telemetry data into a format that is digestible by machine learning anomaly detection techniques?*

In Section 5.1 six key pre-processing steps were implemented in a *dataset tool*:

D1. Dataset splitting — The full span of data was split into separate, training, validation and test sets.

D2. Collation — The varying sample sizes and irregular acquisition times, as well as missing data mentioned in **RQS1** are converted to regular time intervals.

D3. Adding auxiliary data — Auxiliary data sources such as eclipses and orbital data were combined with the primary source for telemetry data.

D4. Scaling — The varying magnitudes of the telemetry data require rescaling to ensure compatibility with deep learning-based forecasting models.

D5. Handling modes — Modes are transformed to a one-hot encoding to ensure compatibility with machine learning models.

D6. Removing anomalies from training data — Semi-supervised models, such as the forecasters used in this thesis, nominally require training data to be free of anomalies. Two methods: interpolation and sample weights were considered, with the latter ultimately chosen. The anomalies themselves were not incorporated into the sample weights, which is marked as an area for improvement in **RQ1S7**.

RQ1 sub-question 3 (RQ1S3)

> *What format and data structure should a dataset of anomalies have to be used as a benchmark?*

As discussed in Section 5.4, the anomalies should be stored in two formats:

- A mask format with the dimensions of the telemetry data (time, channels) and a true or false value depending on if there is an anomaly at that time and in that channel.
- A tabular format where each row is an anomaly with an id, start and end times, and a number of metrics such as whether an anomaly is multivariate or contains missing data.

Additional masks should be created for eclipses, missing data and other rare nominal events. The table and masks should then be accompanied by the telemetry data, completing the benchmark.

RQ1 sub-question 4 (RQ1S4)

> *What anomalies can be found using existing ML-based TSAD techniques?*

The pipeline was run on a number of target channels and successfully obtained detections on three of them. Most of the 40 total detections, presented in Section 6.1, are large anomalies that prominently extend beyond the nominal regime of their respective channels. Such detections are less interesting than the few inconspicuous ones discovered, which would otherwise not be detected using out-of-limits techniques. The result can partly be attributed to the channels analysed: the first has a straight line as its nominal behaviour and the other two are discretised, providing little opportunity for such anomalies. At the same time, the scoring system used to perform the detections is inherently better at finding larger anomalies. More channels should be analysed with the pipeline to confirm its capabilities.

RQ1 sub-question 5 (RQ1S5)

> *How do detected anomalies compare to existing anomalies reported by operators and instrument engineers?*

A collection of anomaly reports was analysed manually, finding seven that coincide with a change in behaviour in the telemetry channels analysed. The subsequent comparison in Section 6.2.1, found that three of these were detected using the pipeline, with the remaining being undetected for various reasons. In discussion with an instrument engineer, two other detections were confirmed to be anomalies. Even though the remaining detections could not be discussed, the two confirmations serve as a good indication of the validity of the results found with the pipeline.

RQ1 sub-question 6 (RQ1S6)

> *What are the differences in approach compared to ESA-ADB?*

The approach presented in this thesis has many similarities in the areas of telemetry data pre-processing, detection post-processing and in the use of semi-supervised forecasting. There are three major differences: the curation of a larger set of initial anomalies before running switching to semi-supervised detection, a greater focus on iteration based on discussion with spacecraft operators, and the manual refinement of detections to ensure correct ground truth. All three aspects are recommended in the updated approach in Section 7.2 and are discussed further in Section 7.1.9 and Section 7.1.10.

RQ1 sub-question 7 (RQ1S7)

> *What are the limitations and flaws of the approach used and what improvements can be made?*

Both the pipeline and methodology developed in this thesis are the first step to a few larger goals and many areas of improvement have been discussed in Section 7.1. These range from small bugfixes to major additions to the approach. For brevity, only the most pressing items are repeated below.

Chief amongst these is the scoring and detection system, discussed in Section 7.1.5, which currently performs poorly on small inconspicuous anomalies, is highly parametrised, and uses a static thresholding system. Improvements could be made by performing automated tuning with a larger collection of pre-labelled anomalies and by implementing a dynamic scoring method as is performed by other works in the field. Alternatively, an improved approach that takes the physical and statistical context of channel analysed could also be considered.

Sample weights were found to be an effective way to ignore anomalous data in forecaster training. Currently eclipses and missing data are weighted zero, but this should be expanded to include known and newly detected anomalies in the future.

Tuning the scoring system and improved use of sample weights requires a larger set of anomalies to be curated. The collection of anomaly reports used in this thesis only contain the most serious anomalies and the addition of other sources should be considered. Alternatively, a first pass with unsupervised anomaly detection methods, which do not require anomaly-free training data, could also yield results. Two such options are described in Section 7.1.7.

More thought should be given to the choice of target and covariate channels which are currently selected without an effective systematic approach. Additionally, the process of applying semi-supervised learning requires iteration to catch and remove all anomalies from training data. Collaboration with instrument engineers or spacecraft operators can significantly accelerate both processes.

To proceed towards an XMM-Newton Anomaly benchmark, an updated approach has been presented in Section 7.2.

Second research question (RQ2)

> *RQ2 - How can explainable AI methods be applied to understand a detected anomaly and its origins?*

As explained in the introduction, time constraints meant that these questions were not fully answered. Only an answer to **RQ2S1** is provided here as it was explored while reviewing literature. Nevertheless, multiple approaches are explored in Section 5.5, which can serve to facilitate future work.

RQ2 sub-question 1 (RQ2S1)

> *Which types of explainable AI methods can be applied to TSAD?*

As explained in Section 2.3, Explainable AI comes in three main forms:

1. Interpretable machine learning models — These are models where the internal workings can be understood directly, such as decision trees and linear regression.
2. Model-specific methods — Here efforts are focussed on making a specific model interpretable by accessing its internals.
3. Model-agnostic methods — These are methods that can be applied to any model.

Only model-agnostic methods were considered, as they do not require efforts to be tied to a single model. Explainable AI can provide global interpretability, which provide insights on an entire model, and local interpretability which provide insights on an individual prediction. Finally, methods can be distinguished by the types of data they can be applied on, such as images, text or tabular data.

In the context of time series anomaly detection a method should be used that can explain individual predictions (e.g. time steps) within tabular data. The two most popular methods that meet these requirements are SHapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME). Of these methods, SHAP was selected because it is actively maintained, due to its compatibility for both local and global interpretation, and due to its stability in comparing nearby predictions.

RQ2 sub-question 2 (RQ2S2)

> *Which temporal and inter-channel dependencies can be uncovered when analysing XMM-Newton anomalies with Explainable AI?*

This question remains unanswered as the methods developed were not complete enough to be applied in analysis.

8.2. Final remarks

This thesis marked the first exploration into XMM-Newton instrument telemetry data with machine learning-based anomaly detection techniques. The introduction presented three primary objectives:

OBJ1 — Find anomalies in XMM-Newton telemetry data by applying machine learning-based anomaly detection techniques.

OBJ2 — Facilitate research into spacecraft telemetry anomaly detection by initiating the construction of a dataset of anomalies in XMM-Newton instrument telemetry data.

OBJ3 — Enable the use of machine learning-based anomaly detection techniques in XMM-Newton telemetry by developing a set of software tools for processing and detecting anomalies in raw XMM telemetry data.

Although fully accomplishing each objective proved challenging, significant strides were made in all three areas. The anomaly pipeline successfully found 40 detections on three target channels and number of these were confirmed to be real anomalies. Investigating XMM telemetry has revealed a number of particularly challenging features, and solutions for these have been partially implemented or suggested in the updated approach shown in Section 7.2. Additionally, considerable progress has been made in the development of software tools which can be adopted for continued use in anomaly detection and the development of a benchmark. Tools built for pre-processing raw telemetry data as well as those for cataloguing detected anomalies have a particularly high level of maturity.

Time constraints meant that the second research question related to Explainable AI for interpreting anomalies, could not be pursued fully. Ultimately, a comprehensive study on the subject applied to spacecraft telemetry could be considered for its own thesis-sized research project. Finally, while a number of areas have been found for improvement within the pipeline and methodology, the development of high-quality general-purpose scoring functions could have significant benefits to anomaly detection field in its entirety. Methods could be explored that can be paired to any forecasting model and potentially account for local context and incorporate physics-informed techniques.

References

[1] M. Tafazoli, "A study of on-orbit spacecraft failures," *Acta Astronautica*, vol. 64, no. 2-3, pp. 195–205, 2009.

[2] S. Flegel, J. Bennett, M. Lachut, M. Möckel, and C. Smith, "An analysis of the 2016 hitomi breakup event," *Earth, Planets and Space*, vol. 69, pp. 1–13, 2017.

[3] F. Sellmaier, T. Uhlig, and M. Schmidhuber, *Spacecraft Operations*. Springer, 2022.

[4] S. Fuertes, G. Picart, J.-Y. Tourneret, L. Chaari, A. Ferrari, and C. Richard, "Improving spacecraft health monitoring with automatic anomaly detection techniques," in *14th international conference on space operations*, 2016, p. 2430.

[5] J. Martinez, "New telemetry monitoring paradigm with novelty detection," in *SpaceOps 2012*, 2012.

[6] J. Martinez, "Drmust-a data mining approach for anomaly investigation," in *SpaceOps 2012*, 2012.

[7] *Publications with xmm newton*. [Online]. Available: https://www.cosmos.esa.int/web/xmm-newton/refereed-publications.

[8] A. N. Parmar, D. Heger, L. Metcalfe, R. Muñoz, and N. Schartel, "Xmm-newton operations beyond the 10-year design lifetime," *Astronomische Nachrichten*, vol. 329, pp. 114–117, 2 Feb. 2008, ISSN: 1521-3994. DOI: 10.1002/ASNA.200710925.

[9] *Extended life for esa's science missions*. [Online]. Available: https://sci.esa.int/web/director-desk/-/extended-life-for-esa-s-science-missions.

[10] N. Moral, P. Pilgerstorfer, F. Marino, *et al.*, "Automation of flight dynamics planning for esa's xmm-newton," Oct. 2024.

[11] G. D. Canio, J. Eggleston, J. Fauste, A. M. Palowski, and M. Spada, *Development of an actionable ai roadmap for automating mission operations*, 2023.

[12] P. Gómez, R. D. Vavrek, G. Buenadicha, J. Hoar, S. Kruk, and J. Reerink, "Machine learning-driven anomaly detection and forecasting for euclid space telescope operations," *arXiv preprint arXiv:2411.05596*, 2024.

[13] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 387–395, Jul. 2018. DOI: 10.1145/3219819.3219845.

[14] S. Schmidl, P. Wenig, and T. Papenbrock, "Anomaly detection in time series: A comprehensive evaluation," in *Proceedings of the VLDB Endowment*, vol. 15, VLDB Endowment, 2022, pp. 1779–1797. DOI: 10.14778/3538598.3538602.

[15] R. Wu and E. J. Keogh, "Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 3, pp. 2421–2429, 2021. DOI: 10.1109/TKDE.2021.3112126.

[16] D. Wagner, T. Michels, F. C. Schulz, A. Nair, M. Rudolph, and M. Kloft, "TimeseAD: Benchmarking deep multivariate time-series anomaly detection," *Transactions on Machine Learning Research*, 2023, ISSN: 2835-8856. [Online]. Available: https://openreview.net/forum?id=iMmsCI0JsS.

[17] L. Herrmann, M. Bieber, W. J. Verhagen, F. Cosson, and B. F. Santos, "Unmasking overestimation: A re-evaluation of deep anomaly detection in spacecraft telemetry," *CEAS Space Journal*, vol. 16, no. 2, pp. 225–237, 2024.

[18] K. Kotowski, C. Haskamp, J. Andrzejewski, *et al.*, "European Space Agency Benchmark for Anomaly Detection in Satellite Telemetry," 2024. DOI: 10.48550/arXiv.2406.17826.

[19] S. Cuéllar, M. Santos, F. Alonso, E. Fabregas, and G. Farias, "Explainable anomaly detection in spacecraft telemetry," *Engineering Applications of Artificial Intelligence*, vol. 133, p. 108 083, 2024.

[20] A. Santangelo, R. Madonia, and S. Piraino, "A chronological history of x-ray astronomy missions," in Springer Nature Singapore, 2024, pp. 3–70, ISBN: 9789811969607. DOI: 10.1007/978-981-19-6960-7_147/TABLES/25. [Online]. Available: https://link-springer-com.tudelft.idm.oclc.org/referenceworkentry/10.1007/978-981-19-6960-7_147.

[21] R. Giacconi, H. Gursky, and L. Van Speybroeck, "Observational techniques in x-ray astronomy," *Annual Review of Astronomy and Astrophysics, vol. 6, p. 373*, vol. 6, p. 373, 1968.

[22] M. Santos-Lleo, N. Schartel, H. Tananbaum, W. Tucker, and M. C. Weisskopf, "The first decade of science with chandra and xmm-newton," *Nature 2009 462:7276*, vol. 462, pp. 997–1004, 7276 Dec. 2009, ISSN: 1476-4687. DOI: 10.1038/nature08690. [Online]. Available: https://www.nature.com/articles/nature08690.

[23] B. J. Wilkes, W. Tucker, N. Schartel, and M. Santos-Lleo, "X-ray astronomy comes of age," *Nature 2022 606:7913*, vol. 606, pp. 261–271, 7913 Jun. 2022, ISSN: 1476-4687. DOI: 10.1038/s41586-022-04481-y. [Online]. Available: https://www.nature.com/articles/s41586-022-04481-y.

[24] N. Schartel, R. González-Riestra, P. Kretschmar, *et al.*, "Xmm-newton," in *Handbook of X-ray and Gamma-ray Astrophysics*, C. Bambi and A. Santangelo, Eds. Singapore: Springer Nature Singapore, 2024, pp. 1501–1538, ISBN: 978-981-19-6960-7. DOI: 10.1007/978-981-19-6960-7_41. [Online]. Available: https://doi.org/10.1007/978-981-19-6960-7_41.

[25] M. Pantaleoni, P. Chapman, R. Harris, *et al.*, "Xmm-newton's operational challenge of changing the attitude control to 4 active reaction wheels, after 12 years of routine operations," 2012. DOI: 10.2514/6.2012-1275587. [Online]. Available: http://arc.aiaa.org.

[26] M. G. Kirsch, A. Elfving, R. Kresken, *et al.*, "Extending the lifetime of esa's x-ray observatory xmm-newton," in *SpaceOps 2014 Conference*, 2014, p. 1608.

[27] *European space agency homepage*. [Online]. Available: https://www.esa.int.

[28] B. J. Wilkes and H. Tananbaum, "The chandra x-ray observatory," in *Handbook of X-ray and Gamma-ray Astrophysics*, C. Bambi and A. Santangelo, Eds. Singapore: Springer Nature Singapore, 2024, pp. 1115–1147, ISBN: 978-981-19-6960-7. DOI: 10.1007/978-981-19-6960-7_150. [Online]. Available: https://doi.org/10.1007/978-981-19-6960-7_150.

[29] M. S. Tashiro, "Xrism: X-ray imaging and spectroscopy mission," *International Journal of Modern Physics D*, vol. 31, no. 02, p. 2 230 001, 2022. DOI: 10.1142/S0218271822300014. eprint: https://doi.org/10.1142/S0218271822300014. [Online]. Available: https://doi.org/10.1142/S0218271822300014.

[30] M. Bavdaz, E. Wille, M. Ayre, *et al.*, "Newathena optics technology," in *Optics for EUV, X-Ray, and Gamma-Ray Astronomy XI*, SPIE, vol. 12679, 2023, p. 1 267 902.

[31] *Esa - final three for esa's next medium science mission*. [Online]. Available: https://www.esa.int/Science_Exploration/Space_Science/Final_three_for_ESA_s_next_medium_science_mission.

[32] *Savechandra: Campaign to save chandra from premature cancellation*. [Online]. Available: https://www.savechandra.org/faq.

[33] F. Jansen, D. Lumb, B. Altieri, *et al.*, "Xmm-newton observatory-i. the spacecraft and operations," *Astronomy & Astrophysics*, vol. 365, no. 1, pp. L1–L6, 2001.

[34] A. Elfving, "The attitude and orbit control of xmm," *ESA bulletin*, vol. 100, pp. 50–54, 1999.

[35] D. De Chambure, R. Lainé, K. Van Katwijk, and P. Kletzkine, "Xmm's x-ray telescopes," *ESA bulletin*, vol. 100, pp. 30–42, 1999.

[36] Turner, M. J. L., Abbey, A., Arnaud, M., *et al.*, "The european photon imaging camera on xmm-newton: The mos cameras," *Astronomy & Astrophysics*, vol. 365, no. 1, pp. L27–L35, 2001. DOI: 10.1051/0004-6361:20000087. [Online]. Available: https://doi.org/10.1051/0004-6361:20000087.

[37] Strüder, L., Briel, U., Dennerl, K., *et al.*, "The european photon imaging camera on xmm-newton: The pn-ccd camera *," *Astronomy & Astrophysics*, vol. 365, no. 1, pp. L18–L26, 2001. DOI: 10.1051/0004-6361:20000066. [Online]. Available: https://doi.org/10.1051/0004-6361:20000066.

[38] den Herder, J. W., Brinkman, A. C., Kahn, S. M., *et al.*, "The reflection grating spectrometer on board xmm-newton," *Astronomy & Astrophysics*, vol. 365, no. 1, pp. L7–L17, 2001. DOI: 10.1051/0004-6361:20000058. [Online]. Available: https://doi.org/10.1051/0004-6361:20000058.

[39] Mason, K. O., Breeveld, A., Much, R., *et al.*, "The xmm-newton optical/uv monitor telescope," *Astronomy & Astrophysics*, vol. 365, no. 1, pp. L36–L44, 2001. DOI: 10.1051/0004-6361:20000044. [Online]. Available: https://doi.org/10.1051/0004-6361:20000044.

[40] *Xmm newton users handbook*. [Online]. Available: https://xmm-tools.cosmos.esa.int/external/xmm_user_support/documentation/uhb/XMM_UHB.pdf.

[41] S. Malik, U. Weissmann, T. Godard, M. G. Kirsch, D. Webert, and A. Mcdonald, "Evolution of the eclipse operations concept for esa's x-ray observatory xmm-newton," in *2018 SpaceOps Conference*, 2018, p. 2574.

[42] H. Barré, H. Nye, and G. Janin, “An overview of the xmm observatory system,” *ESA Bulletin*, vol. 100, pp. 15–20, 1999.

[43] T. Godard, U. Weissmann, L. Toma, W. Zur Borg, K. Yeung, and M. G. Kirsch, “Automated on-ground fdir for esa’s xmm-newton mission,” in *14th International Conference on Space Operations*, 2016, p. 2507.

[44] M. Edirimanne, E. Coe, M. G. F. Kirsch, D. Webert, T. Godard, and U. Weissmann, “Multi-mission spacecraft operations by a single operator with minimal impact on science return,” *SpaceOps*, Mar. 2023.

[45] C. C. Aggarwal, “Outlier analysis,” *Outlier Analysis*, 2017. DOI: 10.1007/978-3-319-47578-3.

[46] *Tsb-ad github repository*. [Online]. Available: https://github.com/TheDatumOrg/TSB-AD.

[47] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, “Outlier detection for temporal data: A survey,” *IEEE Transactions on Knowledge and data Engineering*, vol. 26, no. 9, pp. 2250–2267, 2013.

[48] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[49] H. Xu, W. Chen, N. Zhao, *et al.*, “Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications,” in *Proceedings of the 2018 world wide web conference*, 2018, pp. 187–196.

[50] M. Goldstein and A. Dengel, “Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm,” *KI-2012: poster and demo track*, vol. 1, pp. 59–63, 2012.

[51] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” in *2008 eighth ieee international conference on data mining*, IEEE, 2008, pp. 413–422.

[52] A. Garg, W. Zhang, J. Samaran, R. Savitha, and C.-S. Foo, “An evaluation of anomaly detection and diagnosis in multivariate time series,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 6, pp. 2508–2517, 2021.

[53] S. Ioffe, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015.

[54] K. Cabello-Solorzano, I. Ortigosa de Araujo, M. Peña, L. Correia, and A. J. Tallón-Ballesteros, “The impact of data normalization on the accuracy of machine learning algorithms: A comparative analysis,” in *18th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2023)*, P. García Bringas, H. Pérez García, F. J. Martínez de Pisón, *et al.*, Eds., Cham: Springer Nature Switzerland, 2023, pp. 344–353, ISBN: 978-3-031-42536-3.

[55] Q. Liu and J. Paparrizos, “The elephant in the room: Towards a reliable time-series anomaly detection benchmark,” in *NeurIPS 2024*, 2024.

[56] S. Di Mascio, A. Menicucci, E. Gill, G. Furano, and C. Monteleone, “On-board decision making in space with deep neural networks and risc-v vector processors,” *Journal of Aerospace Information Systems*, vol. 18, no. 8, pp. 553–570, 2021.

[57] B. Ruszczak, K. Kotowski, J. Andrzejewski, *et al.*, “Machine learning detects anomalies in ops-sat telemetry,” in *International Conference on Computational Science*, Springer, 2023, pp. 295–306.

[58] J. Thoemel, K. Kanavouras, M. Sachidanand, *et al.*, “Lean demonstration of on-board thermal anomaly detection using machine learning,” *Aerospace*, vol. 11, no. 7, p. 523, 2024.

[59] K. Kotowski, C. Haskamp, B. Ruszczak, J. Andrzejewski, and J. Nalepa, “Annotating large satellite telemetry dataset for esa international ai anomaly detection benchmark,” in *Proceedings of the 2023 conference on Big Data from Space (BiDS’23)–From foresight to impact–6-9 November*, 2023, pp. 341–344.

[60] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, “Explainable ai: A review of machine learning interpretability methods,” *Entropy*, vol. 23, no. 1, p. 18, 2020.

[61] C. Molnar, *Interpretable Machine Learning, A Guide for Making Black Box Models Explainable*, 2nd ed. 2022. [Online]. Available: https://christophm.github.io/interpretable-ml-book.

[62] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., Curran Associates, Inc., 2017, pp. 4765–4774. [Online]. Available: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

[63] M. T. Ribeiro, S. Singh, and C. Guestrin, “”why should i trust you?”: Explaining the predictions of any classifier,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’16, San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 1135–1144, ISBN: 9781450342322. DOI: 10.1145/2939672.2939778. [Online]. Available: https://doi.org/10.1145/2939672.2939778.

[64] *Shap github repository*. [Online]. Available: https://github.com/shap.

[65] V. Navarro, S. del Rio, M. A. Diego, *et al.*, "Esa datalabs: Digital innovation in space science," *Studies in Big Data*, vol. 141, pp. 1–13, 2024, ISSN: 21976511. DOI: 10.1007/978-981-97-0041-7_1/FIGURES/9. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-97-0041-7_1.

[66] J. Herzen, F. LÃ¤ssig, S. G. Piazzetta, *et al.*, "Darts: User-friendly modern machine learning for time series," *Journal of Machine Learning Research*, vol. 23, no. 124, pp. 1–6, 2022. [Online]. Available: http://jmlr.org/papers/v23/21-1177.html.

[67] Y. Zhao, Z. Nasrullah, and Z. Li, "Pyod: A python toolbox for scalable outlier detection," *Journal of Machine Learning Research*, vol. 20, no. 96, pp. 1–7, 2019. [Online]. Available: http://jmlr.org/papers/v20/19-011.html.

[68] R. Valles, *Esa esac: Xmm newton spice kernels*. [Online]. Available: https://spiftp.esac.esa.int/data/SPICE/XMM/kernels/spk/.

[69] *Jpl/ naif: De432s ephemerides spice kernels*. [Online]. Available: https://naif.jpl.nasa.gov/pub/naif/generic_kernels/spk/planets/.

[70] *Naif: Leap seconds spice kernels*. [Online]. Available: https://naif.jpl.nasa.gov/pub/naif/generic_kernels/lsk/.

[71] A. Fienga, P. Deram, V. Viswanathan, *et al.*, *INPOP19a planetary ephemerides*, 2019. [Online]. Available: https://www.imcce.fr/recherche/equipes/asd/inpop/download19a.

[72] *Fundamentals of astrodynamics and applications*. Springer Science & Business Media, 2001, vol. 12.

[73] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[74] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" In *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, 2023, pp. 11 121–11 128.

[75] S. Hochreiter, "Long short-term memory," *Neural Computation MIT-Press*, 1997.

[76] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-beats: Neural basis expansion analysis for interpretable time series forecasting," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=r1ecqn4YwB.

[77] C. Challu, K. G. Olivares, B. N. Oreshkin, F. Garza, M. Mergenthaler-Canseco, and A. Dubrawski, *N-hits: Neural hierarchical interpolation for time series forecasting*, 2022. arXiv: 2201.12886 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2201.12886.

[78] S. Bai, J. Z. Kolter, and V. Koltun, *An empirical evaluation of generic convolutional and recurrent networks for sequence modeling*, 2018. arXiv: 1803.01271 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1803.01271.

[79] A. Das, W. Kong, A. Leach, S. Mathur, R. Sen, and R. Yu, *Long-term forecasting with tide: Time-series dense encoder*, 2024. arXiv: 2304.08424 [stat.ML]. [Online]. Available: https://arxiv.org/abs/2304.08424.

[80] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, *Attention is all you need*, 2023. arXiv: 1706.03762 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1706.03762.

[81] S.-A. Chen, C.-L. Li, N. Yoder, S. O. Arik, and T. Pfister, *Tsmixer: An all-mlp architecture for time series forecasting*, 2023. arXiv: 2303.06053 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2303.06053.

[82] K. Cabello-Solorzano, I. Ortigosa de Araujo, M. Peña, L. Correia, and A. J. Tallón-Ballesteros, "The impact of data normalization on the accuracy of machine learning algorithms: A comparative analysis," in *International conference on soft computing models in industrial and environmental applications*, Springer, 2023, pp. 344–353.

[83] Y. Di, F. Wang, Z. Zhao, Z. Zhai, and X. Chen, "An interpretable graph neural network for real-world satellite power system anomaly detection based on graph filtering," *Expert Systems With Applications*, vol. 254, p. 124 348, 2024. DOI: 10.1016/j.eswa.2024.124348. [Online]. Available: https://doi.org/10.1016/j.eswa.2024.124348.

[84] B. Ruszczak, K. Kotowski, J. Andrzejewski, C. Haskamp, and J. Nalepa, "Oxi: An online tool for visualization and annotation of satellite time series data," *SoftwareX*, vol. 23, p. 101 476, 2023.

A. Additional anomaly context

This appendix presents companion tables, pipeline settings and additional anomaly plots to support Chapter 6. Each of the final targets were run with two LSTM-based models, *LSTM14* and *BM_LSTM*. Their settings are shown in Table A.1. The main parameters of the final dataset configurations are shown in Table A.2, this also includes the final selected model for each run. T0004 and T0005 use a MinMaxScaler to match more closely with earlier images shown in Chapter 5, while F1128 uses RobustScaler, in line with the findings of Section 5.2.3. Test set metrics are shown in Figure A.1, with the model with the lowest median test-set MAE being selected.

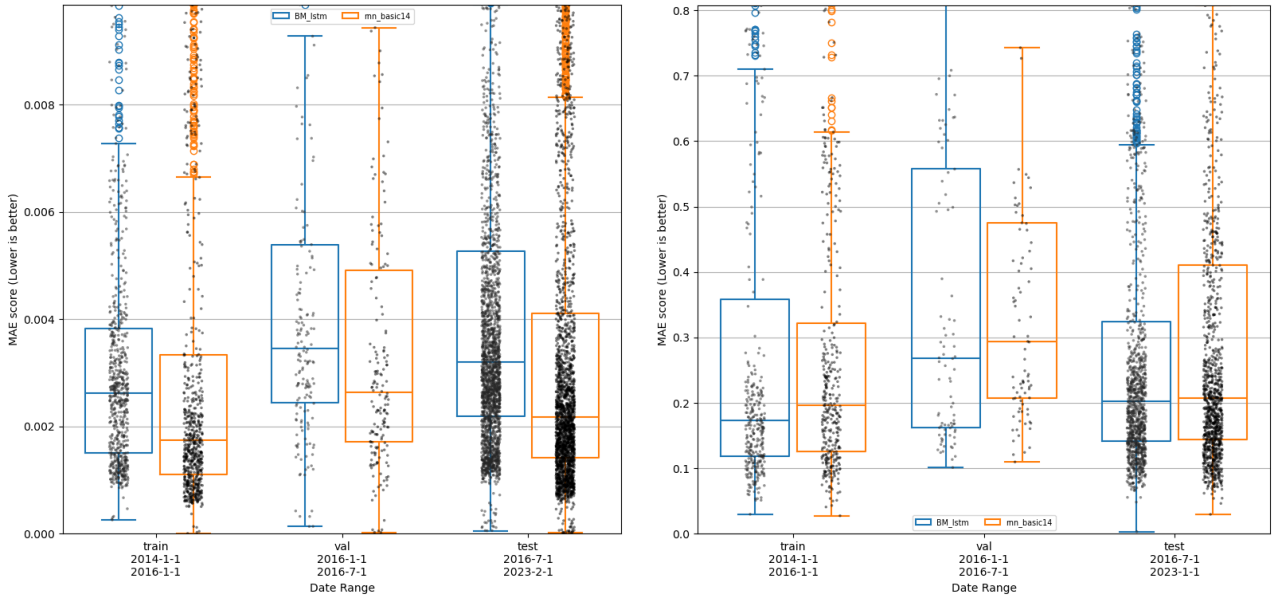
The remaining sections present additional anomaly plots for detections shown in Chapter 6. Appendix A.1 shows an additional anomaly plot for T0004, Appendix A.2 shows multiple for F1128. Finally, the anomaly reports that were not detected in Section 6.2.1, are shown in Appendix A.3.

Table A.1: Hyperparameters for the two forecasting models used to achieve the detections in the results section. All unlisted parameters are set to the defaults as of Darts version 0.32

	BM_LSTM	LSTM14
lookback (input_chunk_length)	12	12
darts model	BlockRNNModel	BlockRNNModel
model	LSTM	LSTM
n_epochs	10	25
n_rnn_layers	1	2
hidden_dim	25	64
batch_size	32	32
dropout	0	0.3
learning rate	0.001	0.0001
loss function	MAE	MAE

Table A.2: Pipeline settings used to achieve the detections shown in the results.

	T0004, T0005	**F1128**	**F1129**
train range:	2014-1-1 to 2016-1-1		
validation range:	2016-1-1 to 2016-7-1		
test range:	2016-7-1 to 2023-2-1	2016-7-1 to 2023-1-1	
target channels:	T0004, T0005	F1128	F1129
covariate channels:	None	F1122, F1198, F1199, F1201, F1190, F1192, F1257, F1258, F1259, F1260, F1191, F1193, F1129	F1122, F1198, F1199, F1201, F1190, F1192, F1257, F1258, F1259, F1260, F1191, F1193, F1128
engineered covariates:	eclipse, distance_earth, distance_sun, beta_angle		
mode_channels:	H5395, H5240, HD013	FD126, FD130	
collation period:	60 seconds		
col. agg. method:	last		
global seed:	1		
scaling:	minmax	robust	
sample weights:	on		
name of model used:	LSTM14	BM_lstm	



(a) T0004 and T0005 (MinMaxScaler)

(b) F1128 (RobustScaler)

Figure A.1: Training metrics for the final two runs. Note that the difference in MAE is due to a difference in scalers used.

A.1. Additional anomaly plots related to T0004 and T0005

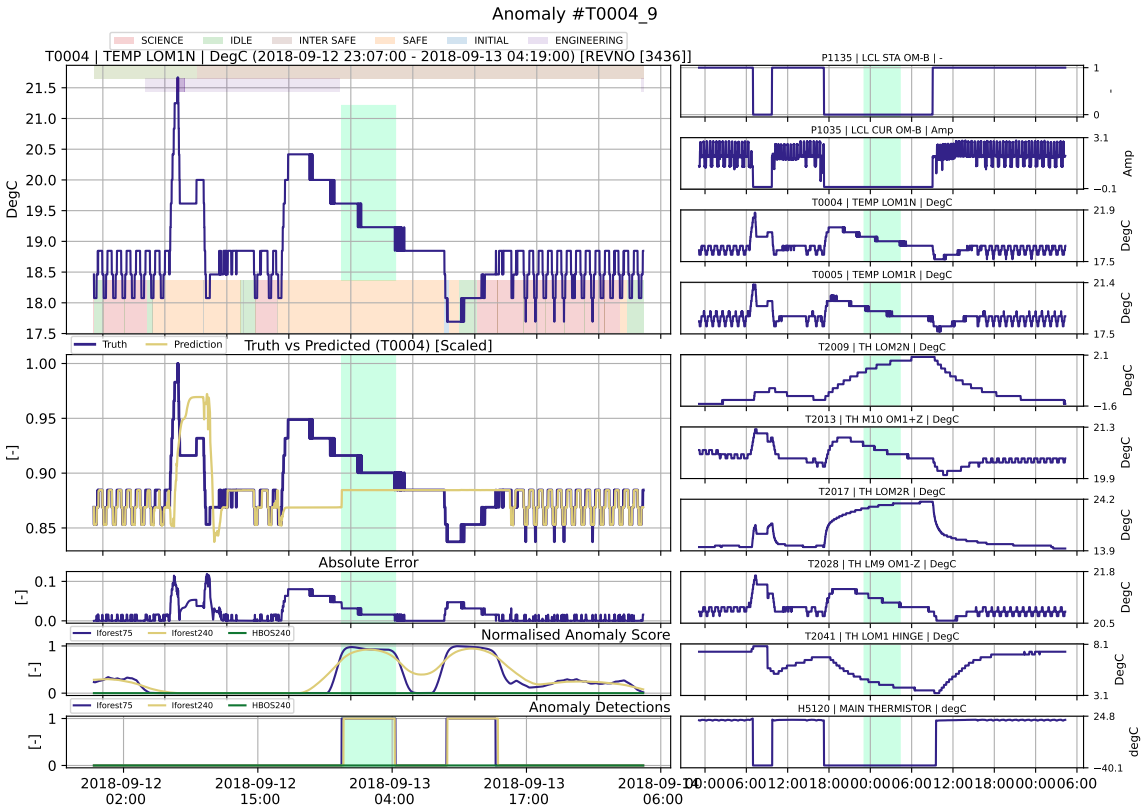


Figure A.2: Full anomaly plot for T0004_73. Related to O2 in Section 6.1.

A.2. Additional anomaly plots related to F1128

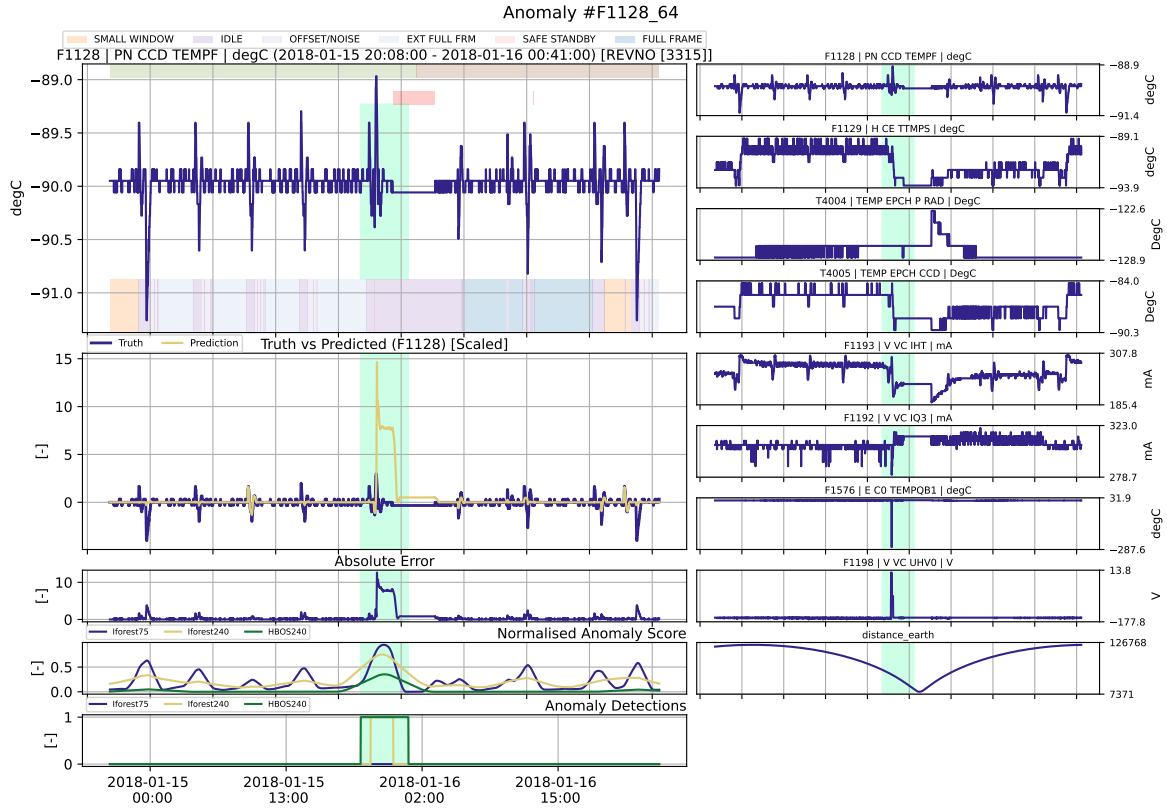


Figure A.3: Full anomaly plot for F1128_64. Related to O5 in Section 6.1.

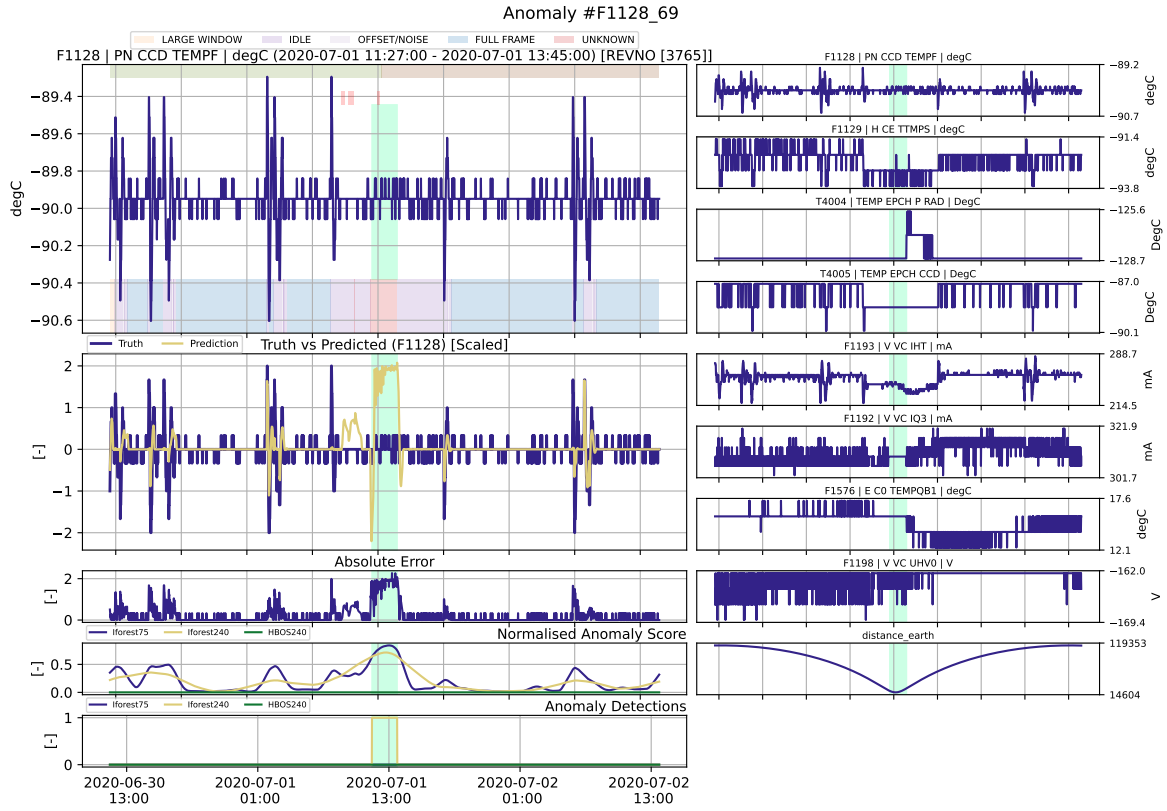


Figure A.4: Full anomaly plot for F1128_69. Related to O5 in Section 6.1.

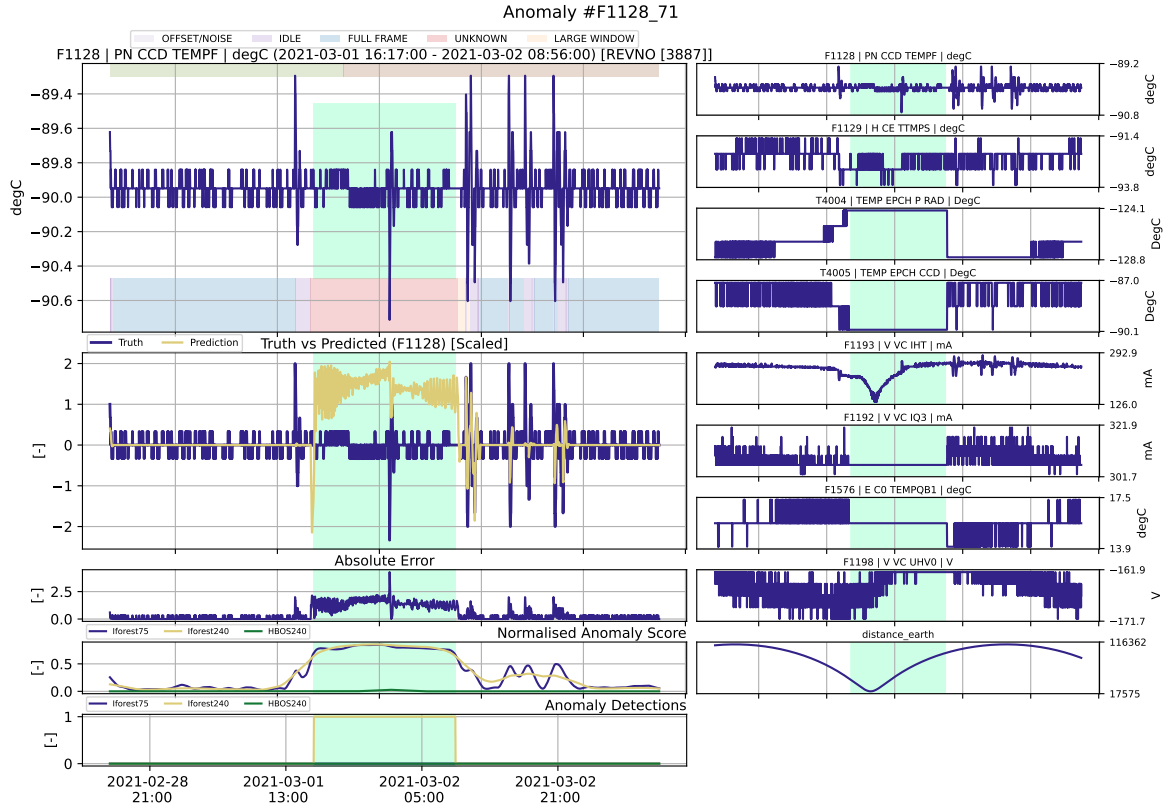


Figure A.5: Full anomaly plot for F1128_71. Related to O5 in Section 6.1.

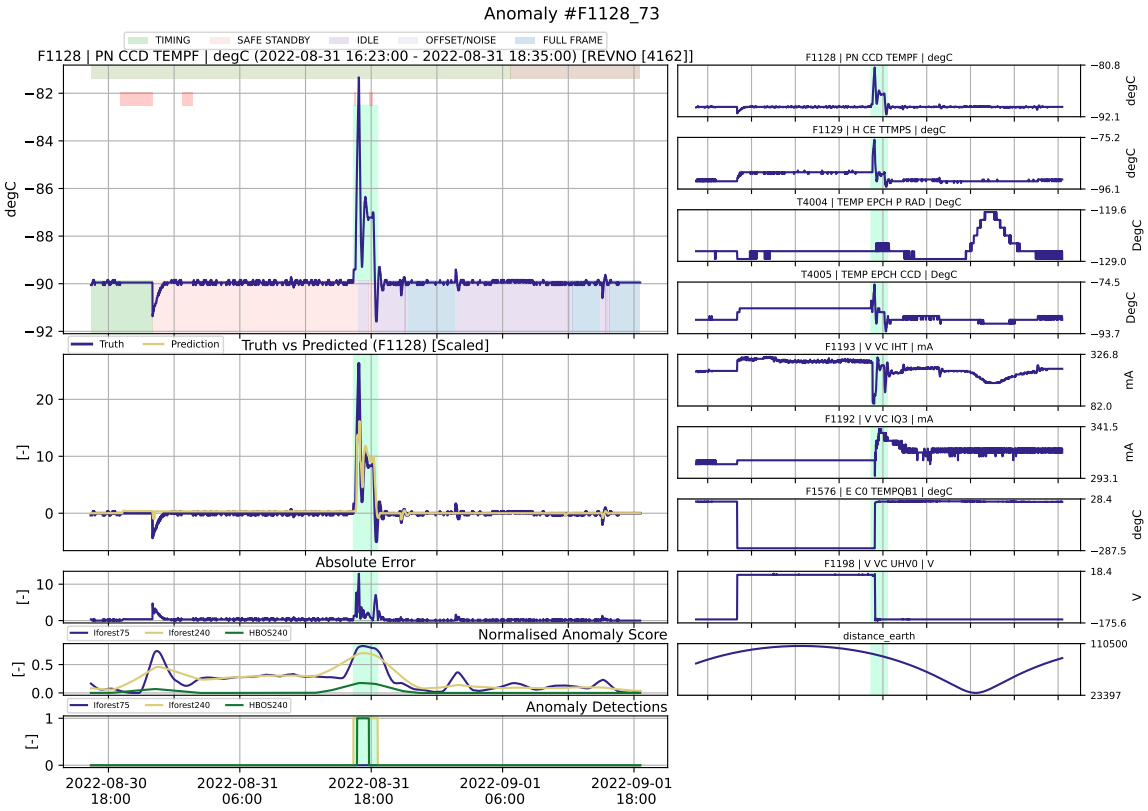


Figure A.6: Full anomaly plot for F1128_73. Related to O5 in Section 6.1.

A.3. Undetected anomaly reports

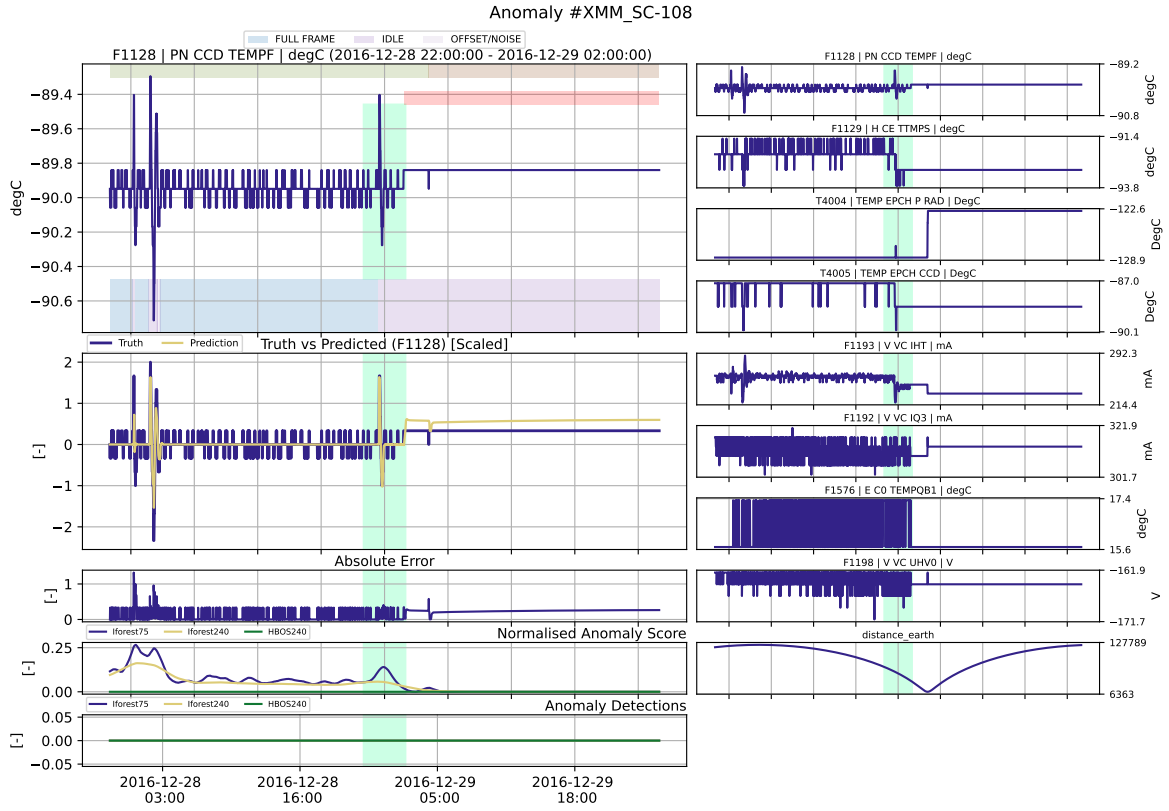


Figure A.7: XMM_SC-108 as represented in F1128. Discussed in Section 6.2.1.

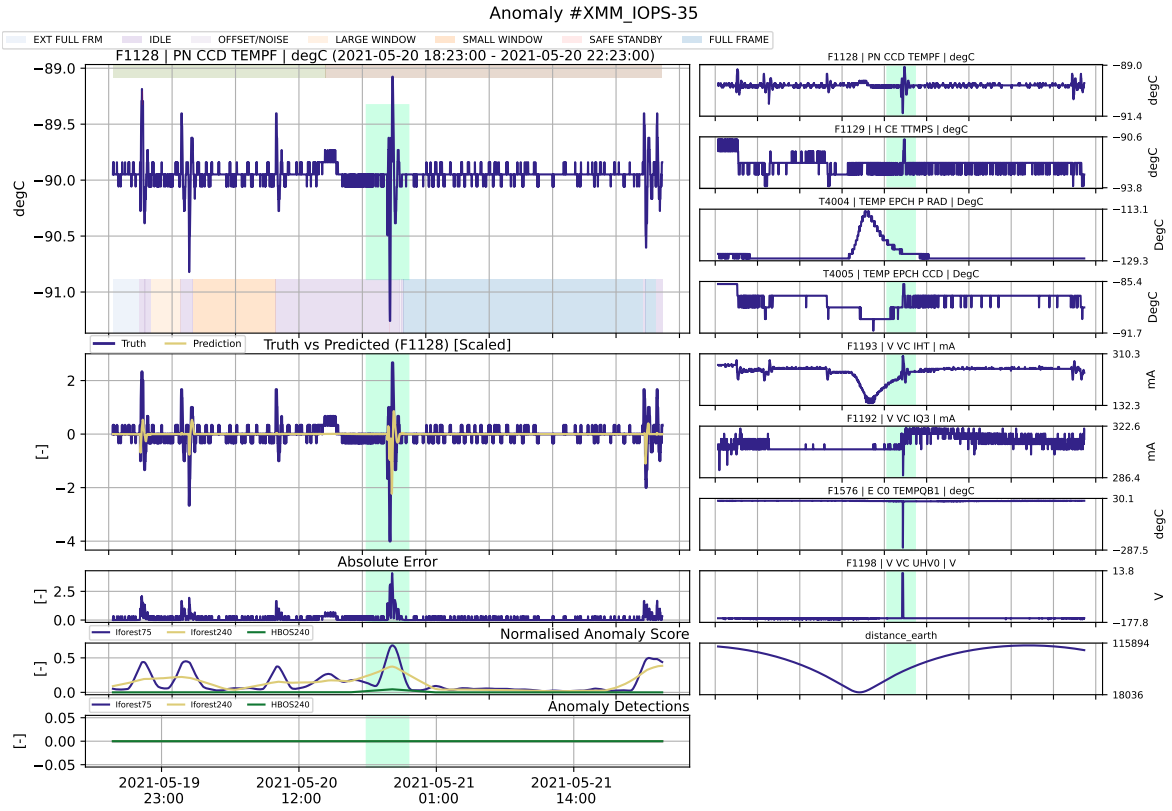


Figure A.8: XMM_IOPS-35 as represented in F1128. Discussed in Section 6.2.1. The problem appears more prominently in other channels.

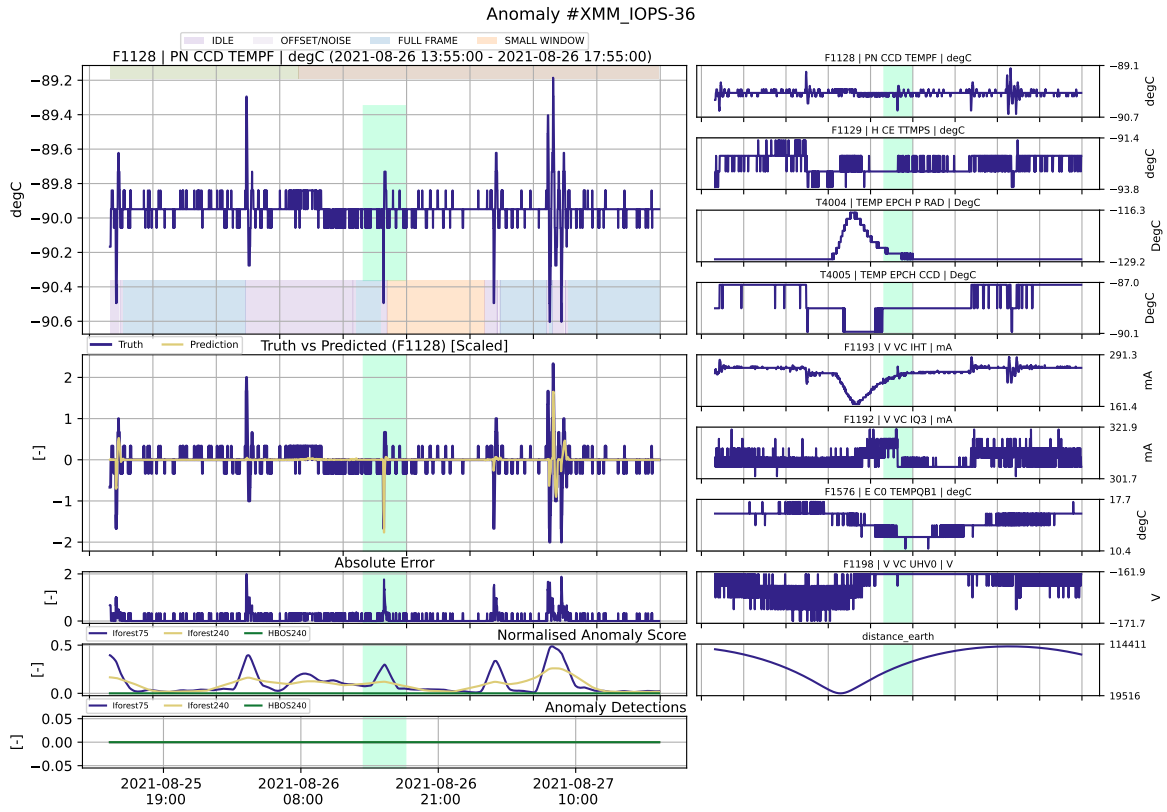


Figure A.9: XMM_IOPS-36 as represented in F1128. Discussed in Section 6.2.1. Surrounding spikes have a greater magnitude than the reported anomaly area.