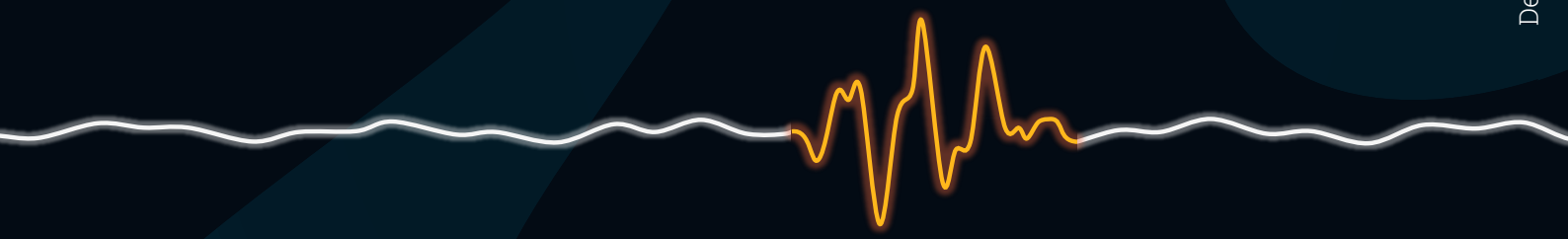
Embedded Spacecraft Fault Detection

A Hitchhiker's Guide to Explainable Thermal Anomaly Alerts for Downlink-Constrained Space Missions

Austin Phillips

Delft University of Technology



TU Delft

This page is intentionally left blank

TU Delft

Embedded Spacecraft Fault Detection

A Hitchhiker's Guide to Explainable Thermal Anomaly Alerts for Downlink-Constrained Space Missions

by

Austin Phillips

to obtain the degree of Master of Science in Aerospace Engineering
and the degree of Master of Science in Embedded Systems
at the Delft University of Technology.

Project duration:	Sept, 2025 – June, 2026	
Thesis supervision:	Dr. S. Speretta,	TU Delft
	Dr. Q. Wang,	TU Delft

Cover: Own work, stylized spacecraft telemetry trace with a localized anomaly segment

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Preface

This thesis marks the closing chapter of my double Master's degree in Aerospace Engineering and Embedded Systems at TU Delft. It feels both very long ago and strangely recent that I walked into my first Bachelor lecture in Delft just under seven years ago.

Even now, I am not sure I could fully explain why I chose Aerospace Engineering in the first place. It was not a decision I understood completely at the time. Looking back, however, I can see that my path was shaped slowly and surely by the people I met, the experiences I gained, and the years I spent living and studying here. For that, I am grateful to God. Delft gave me much more than an education, even if it also made me realize how much I miss the mountains...

First and foremost, I want to thank my parents for their constant love and support, even from more than 7000 km away. I would not be here without you, and I am grateful from the bottom of my heart for everything you have done for me throughout this journey. To my brother, Ethan, thank you for always checking in on me, encouraging me, and being there when I needed it. I am also very grateful to Hannie and Rudi for their kindness, support, and encouragement during my time in the Netherlands. Your warmth and generosity have meant a great deal to me.

To my friends, thank you for making these years memorable. From my time at Tumbleweed, to climbing sessions, runs, and dinners at A^2-housing, thank you for the laughter, support, and company along the way. A special thank you to Alessandro and Lucas for being such an important part of this chapter.

I would also like to thank my supervisors, Stefano Speretta and Qing Wang, for their guidance, feedback, and support throughout this thesis. Thank you for helping me bring the project to the finish line. Finally, I would like to thank the other members of my assessment committee, Erwin Mooij and Alex Caon, for their time and effort in reading and evaluating this work.

And, in the spirit of the Hitchhiker's Guide: Don't Panic.

Austin Phillips
Delft, June 2026

Summary

Small satellites increasingly need to maintain spacecraft health awareness under limited downlink, power, memory, and on-board computing resources. For downlink-constrained missions such as Delfi Twin, the operational problem is not only whether anomalous telemetry can be detected offline, but whether suspected events can be identified on board, summarized compactly, and reported with enough evidence to support initial ground-operator triage. This thesis develops and evaluates an explainable, deployment-oriented, on-board temperature Anomaly Detection (AD) pipeline for a Microcontroller Unit (MCU)-class small-spacecraft setting.

The thesis treats AD as an end-to-end deployment problem rather than as a standalone algorithm-selection task. It defines the mission role, responsibility split, telemetry scope, anomaly semantics, synthetic evaluation protocol, residual-to-alert decision logic, alert-packet content, embedded implementation boundary, and operational claim limits required for a credible prototype. Delfi Twin is used as the case-study mission, with the on-board function scoped as advisory early awareness. The detector may identify unexpected thermal behaviour and package compact evidence, but diagnosis, response selection, and model or procedure updates remain ground-side responsibilities.

The contribution is therefore not a new general-purpose AD algorithm in isolation. Rather, the contribution lies in selecting, combining, implementing, evaluating, and bounding established primitives as a coherent residual-to-alert workflow for explainable, event-level thermal anomaly awareness on MCU-class spacecraft hardware.

The selected pipeline focuses on battery, panel, and MCU temperature telemetry. A lightweight expected-temperature predictor models dominant nominal thermal behaviour, including orbit-scale structure and slow annual variation. The detector then operates on signed residuals between observed and predicted temperature. Residual evidence is converted into bounded events using normalized residual thresholds, two-sided cumulative evidence, persistence, hysteresis, quiet reset, transient-spike suppression, and explicit gap termination. The resulting alert packets are detector-faithful rather than diagnostic: they report the affected node, event timing, deviation direction, residual magnitude, triggering evidence, and clearance reason, without claiming physical root cause on board.

A controlled synthetic benchmark was constructed because available real spacecraft telemetry does not provide the event-level labels needed for quantitative evaluation. The benchmark separates benign telemetry nuisances, telemetry-quality anomalies, and physical thermal-fault families, including heater-control failure, component self-heating, and battery over-temperature or incipient thermal runaway. Under matched-predictor conditions, the final detector recovered all alert-worthy events in the evaluation split, detecting 36/36 telemetry-quality and thermal-fault truth events. The evaluation also showed that event lifecycle logic is necessary: simple thresholding produced substantially more fragmentation and nuisance escalation, while a causal adaptive-limits baseline could not provide the same trade-off between alert-worthy event recovery and alert burden.

The selected pipeline was implemented on an STM32L476RG development board as an STM32L4-class embedded prototype. Host-to-STM32 replay was used to exercise event-centred windows and validation streams while preserving causal on-board execution. The fixed and adaptive implementations processed samples with mean execution times of approximately 0.586 ms to 0.600 ms/sample, with a worst observed processing time of approximately 12.8 ms. This is far below the assumed 15 s telemetry cadence and 60 s alert-frame target. The firmware footprint was approximately 55.88 kB to 57.46 kB flash and 3.33 kB to 3.42 kB RAM, indicating that the detector core is feasible on STM32L4-class hardware at prototype level.

The main deployment condition is predictor validity. The matched-predictor results show that the residual-to-alert detector can recover controlled alert-worthy events when the expected-temperature model remains aligned with the telemetry. Predictor-mismatch experiments showed, however, that small sustained offsets can dominate the residual stream and drive cumulative evidence into persistent false-alert states. This is not a failure of cumulative evidence itself; it reflects the dependence of residual detection on a valid expected-temperature reference. Predictor validity is therefore part of detector validity.

Lightweight adaptive correction was evaluated as a possible mitigation for predictor mismatch. Node-wise and hybrid correction reduced mismatch-driven false-alert occupancy to near-baseline levels in the tested validation scenarios while preserving alert-worthy recall. Targeted embedded replays confirmed that node-wise adaptive correction was implemented, observable, causal, and functionally active on the STM32L4 prototype. The adaptive path re-centred a persistent battery local offset, tracked slow local drift, and suppressed updates during large fault-like residual evidence. However, adaptation also introduces risk: local correction can absorb fault-like behaviour if it is not bounded, gated, frozen during anomaly evidence, and visible to operators.

Real spacecraft telemetry was used as qualitative stress evidence rather than labelled validation. Data from the FUNcube-1 CubeSat provided real examples of orbit-scale thermal structure, panel ripple, telemetry-quality artifacts, contextual thermal variation, and contact gaps. These cases showed that the prediction, residual, evidence, gap handling, and event state workflow remains inspectable on non-ideal on-orbit data. However, they do not provide labelled recall, precision, false alert rate, or confirmed root cause validation for Delfi Twin fault classes. The synthetic benchmark and FUNcube stress tests therefore support different claims: controlled event-level evaluation in the benchmark, and real telemetry workflow plausibility in the FUNcube cases.

Taken together, the evidence in this thesis supports a deployment-oriented prototype claim: an explainable residual-to-alert thermal AD can be implemented with ample timing and memory margins on STM32L4-class hardware, recover controlled alert-worthy events under matched-predictor conditions, and produce compact detector-faithful event alerts for downlink-constrained missions. The prototype is suitable for continued engineering development and hardware-in-the-loop validation. It does not yet establish flight performance, real-fault statistics, complete fault coverage, or autonomous beacon-level operational trust. The remaining challenge is not whether the MCU can execute the algorithm, but whether predictor context, adaptive-correction policy, flight-software integration, and operator workflow can be validated well enough for operational use.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

Abbreviation	Definition
AD	Anomaly Detection
ADCS	Attitude Determination and Control System
ARMAX	Autoregressive Moving Average With EXogenous Inputs
ARX	Autoregressive Model With EXogenous Inputs
CAN	Controller Area Network Bus
CDR	Critical Design Review
COMMS	Communications
COND	Thermal Interface Or Conductivity Degradation
CUSUM	Cumulative Sum
DEG	Surface Or Coating Degradation
EOL	End of Life
EPS	Electrical Power System
ESA	European Space Agency
EWMA	Exponentially Weighted Moving Average
FDIR	Fault Detection, Isolation and Recovery
FeRAM	Ferroelectric Random-Access Memory
FM	Fault Management
FMECA	Failure Modes, Effects, and Criticality Analysis
FSW	Flight Software
FTA	Fault Tree Analysis
GNSS	Global Navigation Satellite System
GPU	Graphics Processing Unit
HEAT	Heater Control Failure
I2C	Inter-Integrated Circuit Bus
IoT	Internet of Things
LEO	Low Earth Orbit
Li-ion	Lithium-Ion
LTAN	Local Time of the Ascending Node
MAE	Mean Absolute Error
MCU	Microcontroller Unit
MET	Mission Elapsed Time
ML	Machine Learning
MODE	Under/Excess Subsystem Self-Heating
MSc	Master of Science
MSL	Mars Science Laboratory
MSP432	Texas Instruments MSP432 Microcontroller
NaN	Not A Number
OBC	On-Board Computer
OBDP	On-Board Data Processing

Abbreviation	Definition
OPS-SAT	Operations Satellite
RAAN	Right Ascension of the Ascending Node
RAM	Random-Access Memory
RMSE	Root-Mean-Square Error
RUN	Battery Over-Temperature Or Incipient Thermal Runaway
SMAP	Soil Moisture Active Passive
SPC	Statistical Process Control
SPI	Serial Peripheral Interface Bus
SPIN	Spin, Attitude, Or Illumination Anomaly
SRAM	Static Random-Access Memory
SSO	Sun-Synchronous Orbit
STM32	STMicroelectronics STM32 Microcontroller Family
STM32L4	STMicroelectronics STM32L4 Microcontroller Family
STM32L476RG	STMicroelectronics STM32L476RG Microcontroller
STM32L496	STMicroelectronics STM32L496 Microcontroller
TIM	Thermal Interface Material
TinyML	Tiny Machine Learning
TLE	Two-Line Element Set
TTA	Time to Awareness
TTC	Time to Criticality
UART	Universal Asynchronous Receiver–Transmitter
UV	Ultraviolet
WCET	Worst-Case Execution Time
XMM-Newton	X-Ray Multi-Mirror Mission

Symbols

Latin symbols

Symbol	Definition	Unit
$A_{\text{ann},j}$	Channel-specific annual temperature modulation amplitude	[°C]
$a_{j,t}$	Sample-level candidate-alarm flag for thermal channel j at sample t	[–]
$A_{\text{spin},j}$	Channel-specific spin-induced temperature ripple amplitude	[°C]
B_{alert}	Size of one compact alert packet	[B]
$b_{\text{common}}(t)$	Common adaptive bias correction shared across channels	[°C]
$b_j(t)$	Applied adaptive bias correction for thermal channel j	[°C]
bias_j	Mean residual bias for thermal channel j	[°C]
$b_{\text{local},j}(t)$	Local adaptive bias correction for thermal channel j	[°C]
B_{queue}	Alert-buffer capacity required for queued alert packets	[B]
$C^-_{j,t}$	Negative cumulative evidence for colder-than-predicted residuals	[–]
$C^+_{j,t}$	Positive cumulative evidence for hotter-than-predicted residuals	[–]
$\Delta T_{\text{ann},j}$	Annual irradiance temperature contribution for channel j	[°C]
$\Delta T_{\text{inj},j}$	Injected temperature deviation applied to thermal channel j	[°C]
$\Delta T_{\text{spin},j}$	Spin-induced temperature ripple for channel j	[°C]
$e_{\oplus}$	Orbital eccentricity of Earth	[–]
$f_{\text{annual},j}$	Node-specific annual modulation term in the thermal predictor	[°C]
f_E	Eclipse fraction of an orbit	[–]

Symbol	Definition	Unit
$f_{\text{stretch},j}$	Node-specific stretched-exponential orbit-scale thermal template	[°C]
h	Orbit altitude	[km]
h_{-}	Negative cumulative-evidence decision threshold	[–]
h_{+}	Positive cumulative-evidence decision threshold	[–]
I	Current	[A]
i	Orbit inclination	[°]
J	Total orbit-template fitting objective	[–]
$J_{\text{Delfi fit}}$	Fitting loss against the Delfi-derived phase-binned temperature profile	[–]
$J_{\text{FUNcube shape}}$	Auxiliary shape penalty based on the normalized FUNcube reference curve	[–]
k_c	Number of quiet samples required to clear an event	[–]
k_{CUSUM}	CUSUM slack parameter	[–]
k_s	Number of candidate-alarm samples required to start an event	[–]
LTAN	Local time of the ascending node	[h]
MAE_j	Mean absolute residual error for thermal channel j	[°C]
n_c	Clear-hysteresis window length	[–]
N_{queue}	Maximum number of alert packets buffered over a blackout interval	[–]
n_s	Start-persistence window length	[–]
R_E	Mean Earth radius	[km]
$\text{RMSE}_{\text{expected}}$	Expected residual spread from unmodelled spin ripple and nominal noise	[°C]
RMSE_j	Root-mean-square residual error for thermal channel j	[°C]
$r_{\odot}$	Earth-Sun distance in astronomical units	[–]
$r_{j,t}$	Temperature residual for thermal channel j at sample t, defined as observed minus predicted temperature	[°C]
$s(t)$	Anomaly score stream as a function of time	[–]
S_{norm}	Normalized annual solar-flux forcing term	[–]
S_{rel}	Relative solar flux due to Earth-Sun distance variation	[–]
s_W	Local variability scale within injection window W	[°C]
T	Temperature	[K]
T_{orbit}	Orbital period	[min]
t_{bus}	Bus transfer latency to the OBC buffer	[s]
t_{compute}	Compute time for detect and diagnose on the OBC (WCET)	[s]
$t_{\text{contact\_wait}}$	Wait time until the next ground contact (store-and-forward delay)	[s]
t_{execute}	On-board execution latency of commanded action	[s]
$t_{\text{gnd\_ingest}}$	Ground ingest latency (downlink to operator pipeline)	[s]
$t_{\text{ops\_decide}}$	Operator or ground-automation decision time	[s]
t_{pack}	Alert pack and encode time	[s]
t_{persist}	Persistence confirmation delay	[s]
t_{preproc}	Preprocessing (group delay) before detection	[s]
t_{sensor}	Sensor and ADC latency	[s]
t_{uplink}	Uplink delivery latency	[s]
$\hat{T}_{\text{base},j}$	Fixed baseline predicted temperature for thermal channel j	[°C]
$T_{\text{clean},j}$	Clean nominal temperature for thermal channel j	[°C]
$\hat{T}_{\text{corr},j}$	Adaptively corrected predicted temperature for thermal channel j	[°C]
t_{days}	Elapsed time in days from the start of the generated year	[d]
$T_{j,t}^{\text{obs}}$	Observed temperature telemetry for thermal channel j, at sample t, after injection	[°C]

Symbol	Definition	Unit
$T_{\text{orbit},j}$	Fitted orbit-scale thermal template for channel j	[°C]
t_p	Approximate day of perihelion used in the annual solar-flux model	[d]
V	Voltage	[V]
$z_{j,t}$	Normalized residual for thermal channel j at sample t	[–]

Greek symbols

Symbol	Definition	Unit
β	Solar beta angle (Sun vector relative to orbital plane)	[°]
δ_s	Solar declination	[°]
ϵ_j	Nominal stochastic temperature-noise term for channel j	[°C]
η_t	Injected ordinary-noise sample at time t	[°C]
κ_t	Spike amplitude factor at time t	[–]
λ_{shape}	Weighting factor for the auxiliary FUNcube shape-support term	[–]
$\mu_{r,j}$	Nominal residual location estimate for thermal channel j	[°C]
Ω	Right ascension of the ascending node (RAAN)	[°]
Ω_s	Right ascension of the Sun	[°]
ω_{spin}	Spacecraft spin rate	[° s^{-1}]
ϕ_j	Channel-specific spin-ripple phase offset	[rad]
ϕ_{spin}	Spin phase	[rad]
π	Circle constant	[–]
σ_{noise}	Standard deviation of injected ordinary telemetry noise	[°C]
$\sigma_{j,\text{nom}}$	Nominal noise standard deviation for channel j	[°C]
$\sigma_{r,j}$	Nominal residual scale estimate for thermal channel j	[°C]
τ_{quiet}	Quiet normalized-residual threshold used for stale-evidence reset	[–]
τ_s	Anomaly-score threshold used for score-to-alarm comparison	[–]
τ_z	Instantaneous normalized-residual threshold	[–]
θ	True anomaly	[°]
θ_E	Eclipse-transition angle used in the orbit-template model	[°]
θ_T	True-anomaly or orbital phase used to evaluate the temperature template	[°]

1

Introduction

CubeSats, and more recently picosatellites, provide low-cost access to space, but their limited downlink, constrained on-board computing, and growing telemetry complexity make traditional, ground-centric Fault Management (FM) increasingly unsustainable [53]. For downlink-constrained space missions, these problems are operational: telemetry backlogs can delay operator awareness of spacecraft health concerns, while continuous full-resolution telemetry downlink uses significant bandwidth. This thesis therefore designs and evaluates an explainable, on-board temperature AD pipeline that produces compact event alerts and operator-relevant context windows on MCU-class hardware.

Delfi Twin is used as a representative downlink-constrained space mission, but the underlying problem is broader. Spacecraft operators often need timely awareness of abnormal behaviour despite limited communication opportunities, delayed telemetry access, and restricted on-board resources. Small satellites provide a useful demonstration setting because these constraints are explicit: the detector must be lightweight, causal, bandwidth-aware, and interpretable enough to support ground-operator triage.

This work is *deployment-oriented*: its focus is not the development of a new general-purpose AD algorithm, but the integration of lightweight, explainable detection into a credible on-board thermal monitoring pathway. It therefore examines how mission role, telemetry observability, anomaly semantics, benchmark construction, anomaly scoring, event formation, alert explainability, and embedded feasibility interact under small-satellite constraints. The literature review and scope analysis in Part I establish the need for this pathway, while the formal research questions, contributions, and deployment-oriented thesis plan are derived in Chapter 5.

1.1. Motivation

As spacecraft telemetry volume and complexity increase, the limiting factor in spacecraft health monitoring is not only whether anomalies can be detected, but whether relevant evidence can reach operators early enough to guide diagnosis, prioritization, and response. Traditional ground-based monitoring depends on telemetry being downlinked, processed, inspected, and interpreted after acquisition. When contact opportunities are intermittent or telemetry is backlogged, abnormal behaviour may be visible in the data before it is visible to operators. An on-board AD function can reduce this delay by identifying suspected events close to the time they occur and prioritizing compact, operator-relevant information for downlink [53].

The operational value of automated AD has already been demonstrated in several spacecraft contexts. Ground-based automation reduced manual telemetry review time for Mars Science Laboratory (MSL) operators from 4 h to 5 h to under 10 min [48]. On-board classification methods such as CloudScout on Φ-Sat-1 show how spacecraft can conserve bandwidth by deciding what information is worth downlinking [26]. Beacon-style operations provide a related example: during New Horizons hibernation, compact beacon-status tones were used to indicate whether the spacecraft was behaving nominally or required operator attention, reducing dependence on continuous housekeeping downlink [73]. Together, these examples motivate a shift from continuous telemetry inspection toward event-centred spacecraft health awareness, where automation reduces review burden, conserves bandwidth, and preserves operator awareness under constrained downlink conditions.

For downlink-constrained space missions, this motivation becomes a practical deployment problem. The detector must operate within limited on-board compute, memory, power, and downlink budgets, while still producing alerts that are meaningful to ground operators. Delfi Twin is used in this thesis as a representative small-satellite case study for evaluating these constraints on MCU-class hardware [54]. A binary anomaly flag is insufficient for this purpose. Operators need to know what changed, when it changed, which telemetry channel was affected, and what detector evidence supported the alert decision. The goal is therefore not only to detect abnormal telemetry, but to convert telemetry deviations into bounded, explainable events that are compact enough for constrained downlink.

Why Temperature?

Temperature telemetry is selected as the first medium for this deployment study because it is physically meaningful, commonly available in spacecraft housekeeping data, and shaped by recurring orbital and operational structure [29, 32, 41]. Temperature deviations may indicate heater-control problems, unexpected subsystem self-heating, battery-related thermal behaviour, sensor or telemetry-quality issues, or longer-term changes in thermal response. The detailed observability argument for temperature telemetry is developed in Chapter 4; here, temperature is introduced as a practical and interpretable case-study signal for evaluating deployable on-board alerting.

1.2. Operational Problem and Mission Need

Despite the potential value of automated AD, developing a suitable and reliable approach for on-board use presents several challenges [21]. First, the pipeline must be feasible on the spacecraft itself. Many AD studies focus on offline performance, but spacecraft commonly impose strict limits on compute time, memory, power, bus usage, and packet size. A detector that performs well offline is not necessarily suitable for flight use.

Second, early anomaly awareness requires the alert timing deadline to be defined explicitly. For a downlink-constrained mission, detection cannot be considered only as an offline classification problem: the spacecraft must process housekeeping telemetry, detect relevant deviations, and prepare compact alert information within the operational timing budget available on board. In this thesis, Delfi Twin provides the representative mission context in which that causal, time-bounded alerting requirement is evaluated.

Third, the output must be explainable enough for ground-operator use. A detector that only reports that an anomaly occurred is of limited operational value. Operators need supporting evidence, including what changed, which channel was affected, how large the deviation was, and for how long it persisted. This is especially important because a temperature deviation may indicate abnormal spacecraft behaviour, but it does not by itself identify the underlying physical cause [32].

Finally, real spacecraft telemetry is imperfect and non-stationary. Telemetry streams can contain gaps, corrupted samples, quantization, calibration changes, mode-dependent behaviour, and mission-phase drift [35]. Component ageing and changing baselines can reduce the validity of nominal thermal models and detection thresholds over the mission lifetime. A deployable AD system must therefore be evaluated not only for nominal benchmark performance, but also for its sensitivity to telemetry artifacts, predictor mismatch, and changing mission conditions.

For downlink-constrained spacecraft, the practical gap is therefore not simply the absence of AD algorithms. The gap is the absence of a demonstrated, deployment-oriented pipeline that connects mission requirements, telemetry suitability, event semantics, explainable alerts, and embedded feasibility on MCU-class hardware. Delfi Twin is used as the case-study mission through which this broader workflow is specified and evaluated.

Operational Problem Statement

There remains a practical gap between the operational need for early, on-board anomaly awareness and what can be executed reliably on MCU-class On-Board Computer (OBC) hardware: real-time processing of temperature telemetry with explainable, operator-useful alerts that remain robust to telemetry artifacts and changing mission conditions.

The following chapters first review the literature and mission context needed to define this deployment problem precisely. The formal research questions and thesis plan are then derived in Chapter 5, after the literature review and scope analysis in Chapters 2 to 4.

1.3. Research Gap and Thesis Positioning

Prior AD work relevant to this thesis can be grouped into two broad categories: method-development studies and deployment-oriented pipeline studies. Method-development studies primarily contribute a new scoring model, or an improvement to an existing model, and are typically evaluated using benchmark datasets and comparative performance metrics. Deployment-oriented pipeline studies instead focus on whether an AD approach can be specified, implemented, explained, and verified for a particular operational use case. This requires more than selecting a scoring method: it also includes the operational role, telemetry assumptions, anomaly definitions, dataset and ground-truth protocol, decision logic, alert structure, and embedded constraints.

This thesis follows the second paradigm. Although spacecraft telemetry AD includes a broad range of methods and comparative evaluations [21], the translation of these methods into mission-aligned, flight-ready capability remains less standardized. Many studies emphasize algorithmic complexity without first establishing whether the available telemetry is suitable for AD, which methods work best for the underlying data, or whether the full pipeline can run within deployment constraints [6, 51]. Accordingly, this thesis prioritizes the design and evaluation of a complete deployable pathway. The selected detector is intentionally lightweight and interpretable. Instead of asking whether a more complex model could achieve higher benchmark performance, this thesis asks whether a suitable end-to-end pipeline can provide mission-useful anomaly awareness under the constraints of Delfi Twin.

Several TU Delft Master of Science (MSc) projects also provide important groundwork for this thesis. Maununen [45] showed that simple Machine Learning (ML) models trained on FUNcube-1 temperature orbits can detect deviations and run on MCU-class hardware, while also highlighting challenges in generalizing across seasonal variation and drift. Castro Traba [69] extended Tiny Machine Learning (TinyML) inference on similar MCU platforms, emphasizing the practical limits of on-device training. Dijkstra [17] investigated explainable AD for spacecraft telemetry using X-ray Multi-Mirror Mission (XMM-Newton) data, but in an off-board setting with Graphics Processing Unit (GPU)-class accelerators. Bhat [5] investigated physics-informed ML for improved temperature prediction, showing performance benefits alongside computational and implementation constraints. Hirs [28] examined grey-box thermal modelling, identifying key challenges for flight deployment.

This thesis builds on those foundations, but differs in emphasis. It integrates lightweight expected-temperature prediction, residual-based scoring, event-forming decision logic, evidence-based alert explanations, synthetic event-level evaluation, real-telemetry stress testing, and MCU-class verification into a single deployment-oriented pipeline. Its contribution is not only in the detector itself, but in a traceable workflow from mission problem formulation to MCU-class implementation and deployment-readiness assessment.

Research Problem Statement

Despite many available AD methods, there is no established deployment-oriented workflow for designing and validating explainable, real-time, on-board temperature AD under MCU-class OBC constraints; this thesis addresses that gap using Delfi Twin as a case study.

The central research problem is therefore not simply whether temperature anomalies can be detected offline. The problem is whether a complete on-board pathway can be defined, implemented, and evaluated: from mission role and telemetry observability, through benchmark construction and residual-event detection, to compact alert reporting and MCU-class verification.

1.4. Research Objective and Questions

The aim of this thesis is provided as follows:

Research Objective

To design, implement, and evaluate a deployment-oriented, explainable, on-board temperature AD pipeline for Delfi Twin as an MCU-class case study, while developing a reusable workflow and evaluation protocol for assessing event-level thermal alerting in downlink-constrained space missions.

This objective is addressed through six research questions, which are previewed here for orientation, and formally derived from the Part I literature review and scope foundation in Chapter 5.

Research Questions

RQ-1: Operational formulation and requirements
What operational role should on-board temperature AD serve in a downlink-constrained mission, and what real-time, responsibility, and resource constraints follow when this role is instantiated for Delfi Twin?

RQ-2: Telemetry suitability and anomaly semantics
Which temperature-telemetry behaviours are observable, nominal, anomalous, and operationally meaningful for event-level AD in the Delfi Twin case-study context?

RQ-3: Evaluation strategy and ground-truth protocol
How can the AD problem be evaluated when real spacecraft telemetry lacks reliable fault-level ground truth?

RQ-4: Pipeline selection and alert formation
What detection pipeline is most suitable for converting temperature telemetry into bounded, compact, and interpretable anomaly alerts under downlink-constrained mission and embedded constraints?

RQ-5: Embedded implementation and performance
Can the selected AD pipeline be implemented on STM32L4-class hardware within the required timing, memory, and alert-packet constraints?

RQ-6: Operational value and deployment readiness
Under what conditions does the prototype provide operational value for downlink-constrained thermal anomaly awareness, and what limitations prevent direct flight deployment or autonomous beacon-style use?

Since these research questions depend directly on the literature review, mission constraints, telemetry observability, anomaly semantics, benchmark strategy, and embedded feasibility requirements, they are derived after Part I. These questions are previewed here to make the thesis argument explicit from the beginning. Chapter 5 derives them formally from the literature review and scope foundation developed in Chapters 2 to 4.

1.5. Thesis Contributions

This thesis makes three deployment-oriented contributions. It does not claim novelty through a new general-purpose AD algorithm; instead, it contributes the mission-specific integration, implementation, evaluation, and claim-bounded interpretation of established AD, event-lifecycle, explainability, and embedded-verification primitives for on-board thermal anomaly alerting.

Thesis Contributions

C-1: Deployment-oriented workflow for event-level thermal alerting
A workflow for assessing explainable on-board thermal anomaly alerting in downlink-constrained missions, instantiated through the Delfi Twin case study.

C-2: Explainable residual-to-event detector and compact alert packets
A lightweight residual-based pipeline that converts expected-temperature deviations into bounded detector events and compact alert packets that preserve the evidence behind each alert.

C-3: Embedded verification and deployment-readiness assessment
An STM32L4-class prototype evaluation that assesses timing, memory use, alert-packet size, robustness, real-telemetry stress behaviour, and the limitations that must be addressed before flight deployment or autonomous beacon-style use.

Together, these contributions define the thesis as an end-to-end deployment study. Chapter 5 clarifies the boundary between established literature primitives and the thesis integration, while Chapter 15 revisits the evidence delivered for each contribution.

1.6. Report Structure

The thesis is organized as follows. Part I reviews the literature and defines the operational, embedded, telemetry, and anomaly scope for deployable on-board AD. Chapter 2 reviews spacecraft FM concepts, telemetry AD pipelines, embedded constraints, telemetry pathologies, ground-truth limitations, anomaly-pattern taxonomies, scoring approaches, decision logic, and explainable alerting. Chapter 3 derives the Delfi Twin case-study mission, including the on-board versus ground responsibility allocation, real-time alerting requirement, latency budget, and embedded resource constraints. Chapter 4 establishes why temperature telemetry is selected, what nominal thermal behaviour must be treated as non-anomalous, which anomaly vocabulary is adopted, and which telemetry pathologies and thermal fault families define the evaluation scope. Chapter 5 then synthesizes these findings into the formal research questions, thesis contributions, and deployment-oriented workflow.

Part II develops the data and method design for the end-to-end pipeline. It constructs the clean nominal thermal reference, creates the labelled synthetic benchmark, screens candidate pipeline architectures, selects and verifies the thermal predictor, and defines the residual-to-alert decision layer. Part III presents the embedded implementation and evaluates the offline and on-hardware behaviour of the selected pipeline. Part IV interprets the operational value of the prototype, identifies limitations and no-go conditions, assesses deployment readiness, and concludes with the main findings and future work.

Part I:

Foundations and Scope for Deployable On-Board Anomaly Detection

This part establishes the problem setting for deployable on-board thermal anomaly detection on Delfi Twin. It defines the operational role, embedded constraints, telemetry observability, anomaly semantics, and scope boundaries that guide the later design decisions.

2

Literature Review and General Research Gap

Part I establishes the literature review and scope foundation for the thesis. The review is selective: it focuses on the literature needed to justify explainable, event-level, temperature-focused, on-board Anomaly Detection (AD) for downlink-constrained Microcontroller Unit (MCU)-class spacecraft. Methods are therefore discussed in terms of their operational role, telemetry assumptions, event-level evaluation requirements, explainability, and embedded feasibility.

The chapter is organized around four literature areas. First, it reviews the operational context of spacecraft AD, including Fault Management (FM) concepts and the role of detection within the wider detect–diagnose–decide–respond chain. Second, it reviews the telemetry and evaluation problem, including spacecraft telemetry imperfections, public dataset limitations, ground-truth challenges, and anomaly-pattern vocabulary. Third, it reviews detection and decision methods, including anomaly-scoring families, expected-temperature prediction, and score-to-event logic. Fourth, it reviews deployment and alerting constraints, including embedded resource limits, compact alert reporting, explainability, and detector-faithful evidence.

Figure 2.1 summarizes this structure. The figure is a map of the literature fields covered in this chapter, not a thesis workflow or implementation architecture.

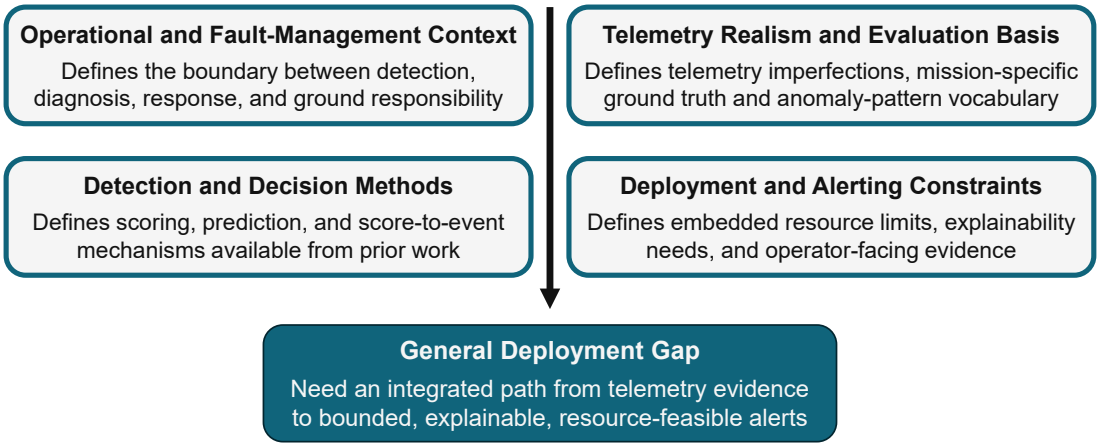


Figure 2.1: Structure of the Chapter 2 literature review. The reviewed fields are grouped by how they contribute to the general deployment gap for on-board spacecraft telemetry AD.

Together, these four areas frame the central gap addressed by the thesis. Spacecraft telemetry AD cannot be assessed by offline score quality alone when the intended use is on board. The detector must operate within a defined FM role, tolerate realistic telemetry imperfections, use a defensible ground-truth strategy, convert evidence into event records, provide compact alert information, and remain feasible under embedded resource constraints.

The following chapters apply this literature foundation to the thesis case study and scope definition. Chapter 3 translates the deployment gap into mission-specific operational and embedded requirements, while Chapter 4 defines the temperature-telemetry and anomaly scope carried forward in the thesis. Chapter 5 then synthesizes the Part I foundation into the research objective, research questions, thesis contributions, and research plan.

2.1. Fault Management and Operational Anomaly Detection

AD is commonly defined as the task of identifying patterns in data that do not conform to expected behaviour [9]. In spacecraft operations, an anomaly is broader than a failure. It is an observed deviation from nominal system behaviour that may reflect a benign artifact, a changing operational context, an incipient fault, or a developing failure [20]. This vocabulary defines the scope of the on-board detector used later in the thesis: it reports unexpected temperature behaviour and supporting evidence, but does not diagnose root cause or decide spacecraft response.

FM provides the wider operational context for AD, and several terms are used throughout this thesis [20]. A *failure* occurs when an intended function is no longer performed acceptably. A *fault* is an internal cause that can lead to such a failure, while a *root cause* is the first fault or environmental condition in the causal chain. An *anomaly* refers to unexpected operational behaviour that may indicate a failure, but may also reflect a benign artifact or changing operating condition. These terms are closely related, but they are not interchangeable.

Operational FM commonly performs detection, diagnosis, decision, response, and adaptation [20]. Detection identifies unexpected behaviour. Diagnosis determines where and what the issue may be. Decision selects an appropriate mitigation. Response executes that mitigation. Adaptation updates thresholds, models, or procedures as knowledge of spacecraft behaviour improves.

This thesis focuses on the detection stage of the FM chain while providing event-level evidence that can support later ground-side diagnosis. The detector can report that a temperature channel was hotter, cooler, or mixed relative to the expected thermal pattern, when the event occurred, how large the residual evidence became, and what evidence triggered or cleared the event. It does not decide whether the physical cause was a heater fault, sensor issue, subsystem self-heating change, battery problem, or another fault. That interpretation remains a ground-operator responsibility.

Real-time relevance in FM is often expressed using the concept of Time to Criticality (TTC). TTC is the time remaining before a fault condition produces an unacceptable mission consequence [20]. In a complete FM chain, detection, diagnosis, decision, and response should occur before that criticality is reached, as shown in Figure 2.2.

A full TTC analysis is outside the scope of this thesis because it would require a detailed Failure Modes, Effects, and Criticality Analysis (FMECA) and Fault tree analysis (FTA) for each relevant failure chain. Different faults can progress toward mission-critical consequences over different timescales, depending on the affected subsystem, propagation path, redundancy, and available mitigations [20]. Chapter 3 therefore derives a narrower and measurable on-board requirement: process native 15 s temperature telemetry and prepare compact alerts within the operational alerting window.

The literature therefore motivates a clear responsibility boundary before detector architecture is selected. On-board AD can identify unexpected telemetry behaviour and provide supporting evidence, while diagnosis, decision-making, response, and model or procedure updates remain broader FM functions. Chapter 3 applies this boundary to the Delfi Twin case study.

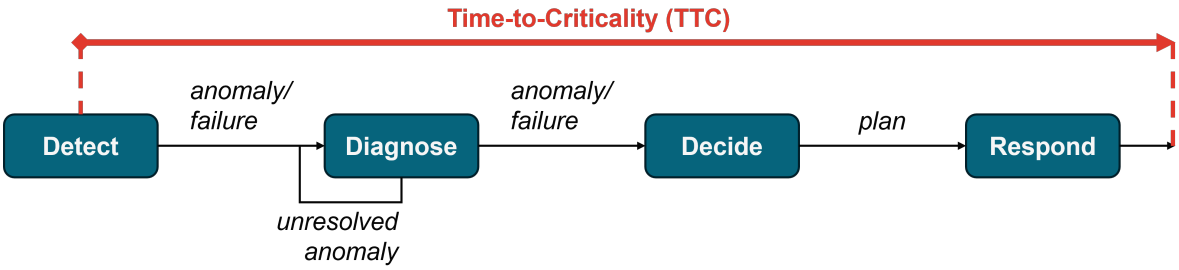


Figure 2.2: Time to Criticality across the Fault Management sequence (adapted from [20])

2.2. Deployment-Oriented Anomaly Detection Pipeline

Most spacecraft telemetry AD systems can be described as a chain from telemetry stream to alert output, as shown in Figure 2.3 and motivated by the diagnostic framework of Bieber et al. [6]. For the present thesis, the useful decomposition is: telemetry acquisition and pre-processing, anomaly scoring, score-to-event decision logic, and compact alert reporting.

Telemetry Stream & Pre-Processing:
The telemetry stream usually consists of time-series measurements from sensors and optional contextual information such as operational modes, status flags, and telecommand activity. Before AD can be applied, the stream may require decoding, calibration, gap handling, resampling, outlier treatment, and alignment of context flags. Spacecraft telemetry is rarely an ideal, uniformly sampled time series, so these pre-processing choices directly affect detector behaviour [35].

Scoring Model and **Decision Logic**
The core detection chain consists of a scoring model followed by decision logic. The scoring model converts telemetry into evidence of deviation from expected behaviour. Depending on the method, this evidence may be expressed as a threshold violation, reconstruction error, distance from nominal examples, low probability under a distribution model, or prediction residual. The decision logic converts this evidence into event records using thresholds, persistence, hysteresis, gap handling, and merge or split rules [9].

Alert Packet:
When an event is declared, the system prepares an alert packet. For operational use, the packet should contain enough detector evidence for ground operators to judge urgency and begin diagnosis, rather than only a binary flag. Useful fields may include affected channel, event timing, residual magnitude, trigger evidence, clearance reason, and references to surrounding telemetry windows for downlink or review. The alert is then interpreted by ground operators within the wider FM process [20, 37].

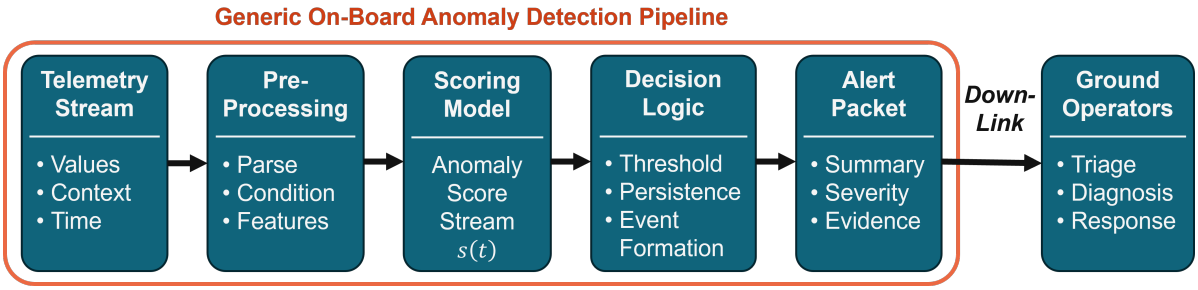


Figure 2.3: Generic deployment-oriented AD pipeline for spacecraft telemetry, from telemetry preprocessing to score generation, event formation, and compact alert reporting.

The scoring and decision stages should therefore be evaluated as a connected score-to-event-to-alert chain. A point-wise anomaly score may be useful for offline benchmarking, but it is not automatically an operational alert.

2.3. Deployment Context for Downlink-Constrained Space Missions

Downlink-constrained missions provide the broader operational setting considered in this thesis. Telemetry may be generated faster than it can be reviewed or downlinked, so value depends not only on detecting abnormal behaviour, but also on identifying which evidence should be prioritized for transmission. This setting is restrictive for small spacecraft, where power, memory, compute, and packet budgets limit what can be performed continuously on board.

Delfi Twin is used later as a representative case study at this constrained end of the design space. The spacecraft build on Delfi-PQ heritage and are expected to operate under tight power, volume, downlink, and computing constraints [54, 64]. The mission-specific Delfi-PQ and Delfi Twin context is introduced in Chapter 3; here, the point is that the literature problem is not ground-based anomaly scoring, but on-board event alerting under intermittent observability and MCU-class resources. The Delfi-PQ satellite and internal subsystem stack are shown in Figure 2.4.

Three deployment consequences follow. First, the telemetry stream available on board can differ substantially from the sparse ground-received samples available for later analysis. Second, nominal behaviour can depend on operating mode, power state, communication activity, attitude, illumination, and thermal context. Third, limited memory, compute, power, bus, and packet budgets restrict which methods can run continuously on board.

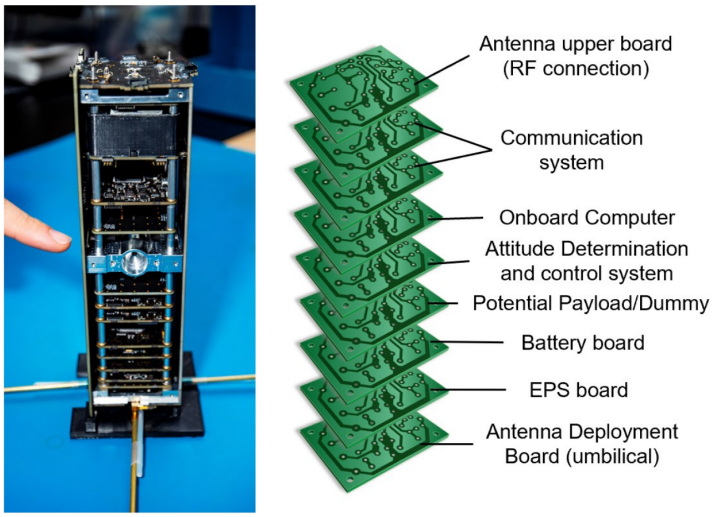


Figure 2.4: Delfi-PQ satellite and subsystem stack (from [45])

This setting motivates methods that are causal, lightweight, interpretable, and event-level. Causality is required because an on-board detector cannot use future samples. Lightweight implementation is required because the detector must coexist with flight-software tasks under limited resources. Interpretability is required because alerts should support operator triage rather than produce opaque flags. Event-level output is required because constrained downlink favours compact summaries and relevant context windows over continuous full-resolution telemetry.

Chapter 3 derives the concrete Delfi Twin operational role, timing requirement, and embedded resource assumptions used later in the thesis evaluation.

2.4. Embedded Constraints for On-Board Anomaly Detection

On-board AD must satisfy constraints that are often secondary in offline AD studies. The most relevant embedded constraints are compute time, Random-Access Memory (RAM) use, flash footprint, power consumption, data bus load, and alert-packet size.

Compute time determines whether the detector can process each telemetry sample within the available timing budget. RAM limits how much recent telemetry, intermediate detector evidence, and event-state information can be stored during operation. Flash limits the size of the flight-software code and any stored model parameters. Power constraints limit how much additional processing can be performed continuously on board: the AD function should therefore have a low computational duty cycle, avoid sustained processor load, and minimize unnecessary data movement [65]. Shared buses can introduce latency and jitter if AD-related data movement becomes too frequent or too large. Alert-packet size also matters because the detector output must fit within limited downlink opportunities.

Reported embedded AD implementations show that these constraints can be satisfied on resource-limited hardware, but the available comparisons are fragmented across domains, platforms, sensor modalities, and output semantics. Maunonen [45] is the closest local precursor: it demonstrated neural-network temperature monitoring on a Delfi-PQ-class low-power On-Board Computer (OBC) using TensorFlow Lite Micro on an MSP432 MCU. This thesis builds on that local context, but shifts the focus from anomaly-probability output to residual-to-event alerting, compact alert packets, predictor-validity analysis, and embedded replay.

Horne et al. [29] provide one of the closest published spacecraft thermal-telemetry implementation references because they report microcontroller timing, memory, and power for a CubeSat temperature AD. Thoemel et al. [68] provide relevant spacecraft thermal-anomaly and in-orbit demonstration context, although their infrared-image payload runs on a Raspberry Pi rather than a low-rate MCU telemetry detector. Other embedded AD studies report resource use for non-space domains such as structural health monitoring, trusted Internet of Things (IoT), and industrial monitoring [46, 56, 1]. These studies motivate treating embedded feasibility as an explicit evaluation dimension, while avoiding direct performance ranking across incomparable platforms and tasks.

Memory technology also affects feasible on-board design. On-chip flash is appropriate for firmware, static constants, and stored model parameters, but frequent small updates are undesirable. Static Random-Access Memory (SRAM) is better suited to volatile working data, such as recent telemetry samples, intermediate detector evidence, and event-state information. Non-volatile memories such as Ferroelectric Random-Access Memory (FeRAM) or external flash can support persistent configuration, queued alert packets, event records, or bulk logs, but their suitability depends on capacity, endurance, access latency, and write behaviour [23]. Feasible on-board design therefore requires attention not only to total memory capacity, but also to correct placement and update behaviour within the memory hierarchy.

For on-board use, a detector should therefore be assessed not only by detection performance, but also by resource use. A method that performs well offline may still be unsuitable on board if it requires long telemetry histories, large working memory, computationally intensive inference, substantial stored parameters, or post-processing that exceeds the available timing, memory, or power budget. Chapter 3 translates these constraints into Delfi Twin prototype budgets, and Chapters 11 and 12 verify the implemented pipeline against measured timing, memory, packet-size, and resource estimates.

2.5. Spacecraft Telemetry Realism and Pathologies

Spacecraft telemetry is rarely an ideal, uniformly sampled time series. Real telemetry can include *pathologies*, such as missing data, irregular timestamps, corrupted samples, quantization, calibration changes, drift, and mode-dependent behaviour [58, 35]. These behaviours can make AD difficult because they may either resemble true physical anomalies or obscure the evidence needed to detect them.

Telemetry gaps are particularly important for event-level detection. During an unobserved interval, the detector cannot know whether an event persisted, cleared, or changed character unless additional context is available. Gap handling should therefore be part of the event-lifecycle design, not merely a missing-data correction step.

Telemetry artifacts also need to be separated from physical faults. An isolated spike may be a benign nuisance. A pinned value or saturation interval may not represent a thermal fault, but it may still be alert-worthy because it compromises interpretation of the temperature stream. Similarly, quantization and low-amplitude measurement noise should normally be tolerated, while abrupt calibration steps or long held values may require operator awareness.

These telemetry pathologies motivate a distinction between telemetry artifacts, telemetry-quality anomalies, and physical spacecraft faults. For the present scope, these categories affect both labelling and event handling: gaps affect event continuity, isolated spikes require nuisance suppression, and pinned values or calibration steps may require operator awareness even when they are not physical thermal faults. Chapter 4 applies this distinction to Delfi Twin temperature telemetry.

2.6. Ground Truth, Public Datasets, and Benchmark Strategies

Evaluating an AD pipeline requires ground truth: labels indicating when anomalous behaviour occurs and what kind of behaviour it represents. Public spacecraft datasets are valuable because they expose realistic telemetry properties, including gaps, irregular sampling, mode changes, multichannel correlations, drift, rare events, and non-ideal measurement behaviour. However, they often do not provide the fault-level labels required for mission-specific evaluation.

2.6.1. Public Spacecraft Datasets and Label Limitations

Most spacecraft AD studies rely on real public or anonymized telemetry datasets [21], such as Soil Moisture Active Passive (SMAP), Mars Science Laboratory (MSL), Operations Satellite (OPS-SAT), Sentinel, X-ray Multi-Mirror Mission (XMM-Newton), and FUNcube-1. These datasets are valuable for studying realistic telemetry behaviour and testing detectors on historical spacecraft data. However, many are unlabelled, and even labelled datasets often identify anomalous intervals without specifying the underlying physical fault, telemetry pathology, or event-level interpretation. Recent operations-oriented benchmarks partially improve this situation by distinguishing anomalies from rare nominal events, communication gaps, and invalid segments, and by incorporating contextual information such as telecommands and non-target channels [35, 21].

Interval-level labels may be sufficient for detection-focused evaluation, but not for validating event-level alerting. Here, validation requires labelled expectations for channel attribution, event timing, anomaly pattern, telemetry-quality status, possible physical interpretation, and alert-worthiness. These labels allow the implementation to be checked for detection correctness, event formation, nuisance suppression, and operator-useful alert content.

2.6.2. FUNcube-1 as a Real-Telemetry Reference Case

FUNcube-1, also known as AO-73, is a 1 U educational CubeSat developed by AMSAT-UK, AMSAT-NL, and ISIS-BV. The mission was designed for educational outreach and amateur-radio use, and includes a telemetry beacon, dashboard software, and a public data warehouse for decoded telemetry received by ground stations [3, 22]. FUNcube-1 is introduced here because it provides public, real small-spacecraft temperature telemetry that can be used to inspect orbit-scale thermal morphology, telemetry gaps, artifacts, and short-timescale thermal ripple.

FUNcube-1 is not used as a labelled validation set. Its spacecraft geometry, orbit, sensor placement, attitude behaviour, thermal design, and operational history differ from Delfi Twin. Direct transfer of temperature values, thermal parameters, fault labels, or detector performance metrics would therefore be inappropriate. Its role is instead to complement the labelled synthetic benchmark: the synthetic data support controlled event-level evaluation, while FUNcube-1 supports real-telemetry morphology checks and qualitative stress testing. The specific roles of FUNcube-1 telemetry are summarized in Table 2.1.

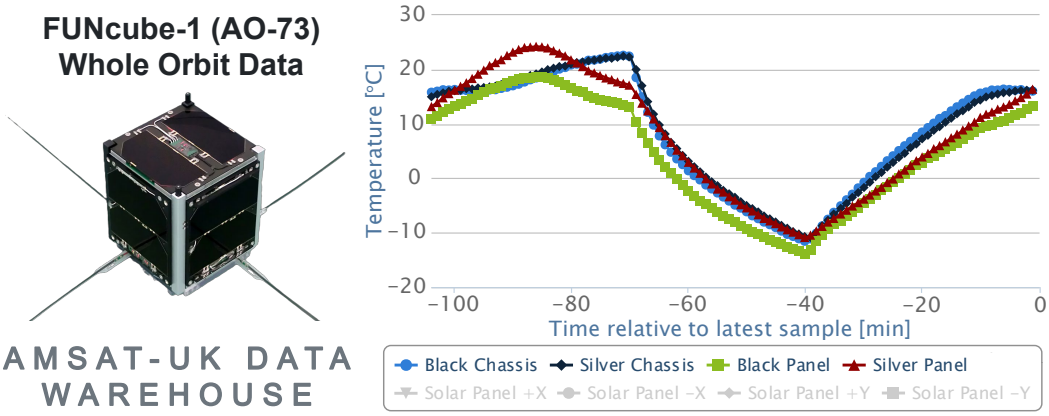


Figure 2.5: Example FUNcube-1 (AO-73) whole-orbit telemetry view, adapted from the AMSAT-UK FUNcube-1 data warehouse [22]

In later chapters, selected FUNcube windows are used to inspect whether the implemented pipeline remains interpretable under real gaps, artifacts, contextual thermal changes, and ripple regimes. Because independent fault labels and complete attitude-state truth are unavailable, detector-declared FUNcube intervals are not interpreted as confirmed faults, true positives, or false positives. This preserves the claim boundary used throughout the thesis: synthetic results measure controlled detector performance, whereas FUNcube results demonstrate real-telemetry interpretability and expose operational complications that must be handled before flight use.

Table 2.1: Role of FUNcube-1 telemetry as a real-telemetry reference and qualitative stress-test source.

Use of FUNcube-1	Evidence added	Claim boundary
Thermal waveform reference	Shows real orbit-scale heating/cooling and panel or frame temperature variation in small-spacecraft telemetry.	Not a Delfi Twin thermal model.
Thermal-shape support	Provides examples of real thermal waveform shapes, partial phase coverage, and contact-gap structure.	Shape support only; no direct temperature or parameter transfer.
Telemetry realism check	Contains real gaps, artifacts, contextual thermal variation, and ripple regimes.	No labelled recall, precision, or false-alert rate.
Qualitative stress-test source	Allows later inspection of residual and event-evidence behaviour on non-ideal real spacecraft telemetry.	Detected events are not treated as confirmed faults.

2.6.3. Constructing a Mission-Specific Evaluation Dataset

Three dataset construction strategies were considered for Delfi Twin thermal AD: manual generation from a thermal model, direct use of existing spacecraft telemetry, and synthetic injection into a nominal baseline. These strategies are summarized in Table 2.2.

Table 2.2: Dataset construction options considered for Delfi Twin thermal AD.

Strategy	Main value	Main limitation	Deployment relevance
Manual generation from a thermal model	Physically interpretable and controllable if a validated spacecraft thermal model exists.	High-fidelity model development is time-intensive and beyond the scope of this prototype.	Used only in simplified form through the nominal thermal truth model.
Existing spacecraft telemetry	Realistic sampling, noise, gaps, operational effects, and long-term behaviour.	Labels are often missing, incomplete, or not fault-specific; telemetry is not Delfi Twin-specific.	Used to inform nominal waveform behaviour and anomaly patterns, not as direct ground truth.
Synthetic injection into a nominal baseline	Provides explicit onset, offset, affected channel, anomaly class, and parameter labels.	Can become unrealistic if not anchored to mission-relevant telemetry and literature.	Suitable when controlled event-level labels are required and real labelled faults are unavailable.

Based on this comparison, synthetic injection into a nominal baseline is a suitable strategy when mission-specific labelled faults are unavailable but controlled event-level evaluation is required. Its strength is label control: onset, offset, affected channel, anomaly class, and parameter values can be specified explicitly. Its weakness is realism: injected events can become unrepresentative if they are not anchored to mission-relevant telemetry behaviour and literature-derived fault morphology.

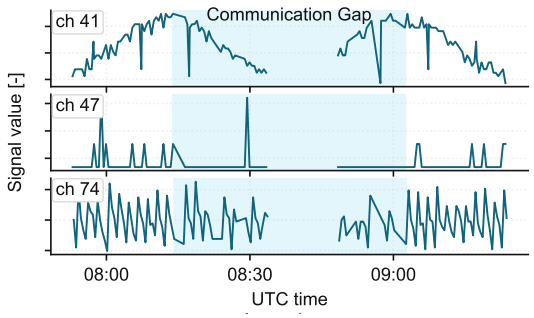
Schefels et al. [60] provide a relevant example of this general strategy through a synthetic satellite telemetry library that generates satellite-like time series and injects labelled anomaly primitives to address the lack of labelled spacecraft telemetry for machine-learning evaluation. This thesis follows the same broad evaluation pattern by using synthetic telemetry to create controlled event-level truth when suitable real flight-fault labels are unavailable. However, the benchmark developed here is mission-specific rather than a direct wrapper around the DLR library. It adds a Delfi Twin-class thermal evaluation layer, including a nominal thermal baseline, affected-node event labels, orbit-aware gap handling, injection metadata, and thermal-fault families required for residual-to-event detector assessment.

This trade-off is central to the evaluation strategy. A synthetic benchmark can support controlled event-level analysis, but it cannot replace future validation against labelled, operator-reviewed, or otherwise validated real telemetry.

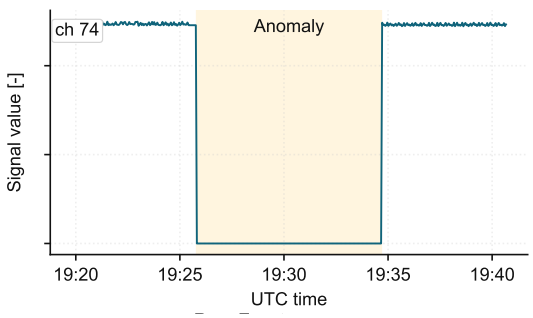
2.6.4. Existing Anomaly Patterns as Design References

Although existing spacecraft datasets are not used as direct ground truth for Delfi Twin, they are useful for identifying realistic anomaly patterns in spacecraft telemetry. These examples do not directly define Delfi Twin thermal faults, but they inform the telemetry pathologies and deviation shapes that should be represented in the synthetic benchmark.

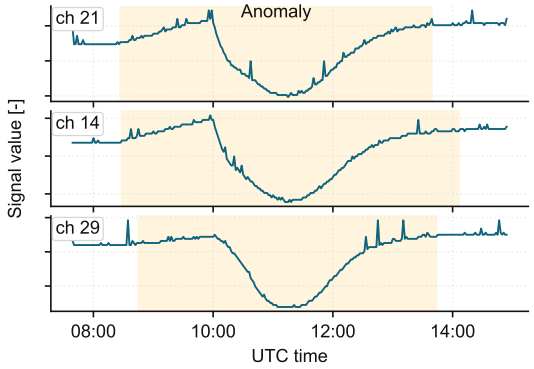
The European Space Agency (ESA) anomaly examples in Figure 2.6 motivate communication-gap, dropout, smooth-excursion, and multichannel-propagation scenarios. The SMAP and MSL examples in Figure 2.7 motivate held-value, staircase-like, point-impulse, and switching-pattern scenarios.



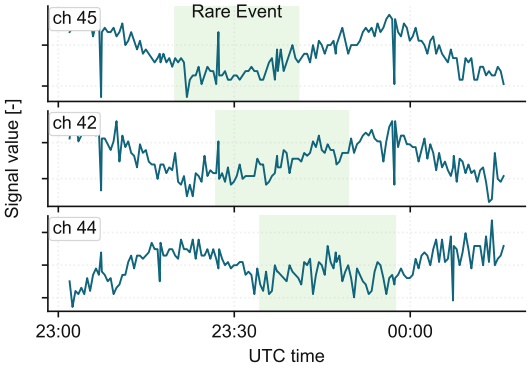
(a) Representative communication-gap event motivating missing-data windows and dropout campaigns.



(b) Discrete dropout event motivating pinned-value or state-dropout anomalies.



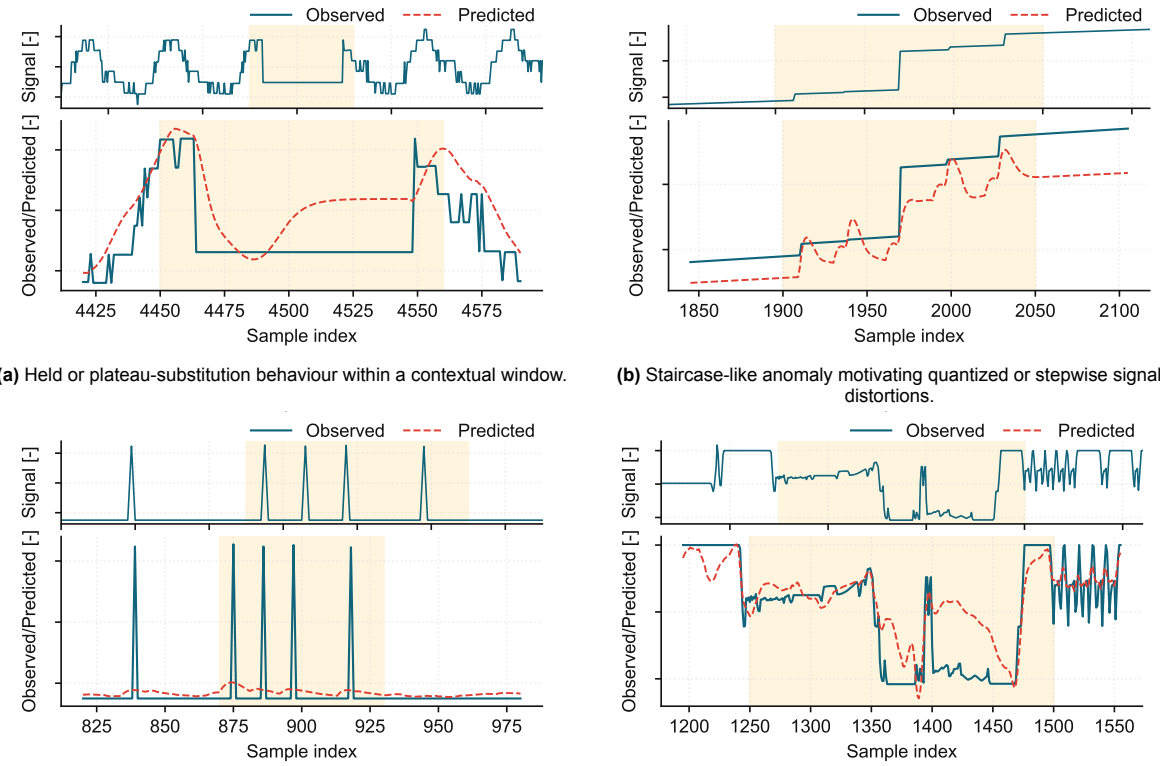
(c) Localized smooth multichannel excursion motivating shift-and-recovery thermal transients.



(d) Coupled multichannel anomaly with staggered onset, motivating delayed multichannel realizations.

Figure 2.6: Representative ESA anomaly examples used to motivate morphology only; not direct Delfi Twin validation. [15]

For this thesis, these anomaly patterns are not labelled validation cases. They are used only as design references for constructing controlled scenarios that test whether the detector can tolerate benign nuisances, detect telemetry-quality anomalies, and form bounded events for physically meaningful thermal deviations.



(a) Held or plateau-substitution behaviour within a contextual window.

(b) Staircase-like anomaly motivating quantized or stepwise signal distortions.

(c) Point-like impulse anomalies motivating isolated single-point or short-transient perturbations.

(d) Multilevel switching regime change motivating state-dependent contextual anomalies.

Figure 2.7: Representative SMAP and MSL anomaly examples used to motivate morphology only; not direct Delfi Twin validation. [50]

These examples establish the claim boundary used later in the thesis. Controlled synthetic benchmarks support event-level metrics such as recall, time-to-awareness, nuisance burden, and false-alert rate. Real spacecraft telemetry supports qualitative realism checks because it exposes detectors to gaps, irregularity, drift, mode changes, and non-ideal measurements. Without event-level truth, it cannot support mission-specific claims about fault recall, false-alert rate, or alert-worthiness.

2.7. Spacecraft Anomaly Pattern Taxonomies

Anomaly-pattern taxonomies provide vocabulary for describing how abnormal behaviour appears in telemetry. General AD literature often distinguishes point, contextual, and collective anomalies [9]. A *point anomaly* is a single data point that is abnormal relative to the rest of the data. A *contextual (conditional) anomaly* is abnormal only under a specific context, such as orbital phase or operational model. A *collective anomaly* is a sequence of related samples that is abnormal as a group, even if the individual samples themselves are not abnormal.

Literature focused specifically on spacecraft telemetry anomaly patterns further distinguishes common morphology types [11]. These include isolated impulses, step changes, drift, contextual shifts, collective excursions, plateaus, and multichannel effects. For temperature telemetry, these morphologies imply different follow-up questions. A single high sample may be a benign spike, a sustained positive residual may suggest excess heating, a negative residual may indicate missing expected heating, and a smooth drift may reflect changing thermal interfaces, seasonal mismatch, or stale predictor parameters. Multichannel excursions can also indicate coupled thermal behaviour or shared environmental context.

This vocabulary is carried forward as the common language for temperature-telemetry scope, synthetic event labels, and detector-faithful alert evidence.

2.8. Anomaly Scoring and Thermal Prediction Families

Spacecraft telemetry AD methods can be grouped by learning paradigm, by whether they analyse individual channels or multivariate telemetry, by modelling assumptions, or by the way they generate anomaly evidence [61]. This thesis focuses on the evidence produced by each method, because the evidence must later support causal event formation, compact explanation, and MCU-class implementation.

This section reviews two connected bodies of literature. Subsection 2.8.1 reviews generic anomaly-scoring families. Subsection 2.8.2 reviews expected-behaviour and thermal prediction models relevant to residual-based detection. The aim is to define the literature basis for the thesis-specific screening performed later.

2.8.1. Scoring Families for Spacecraft Telemetry Anomaly Detection

Sequential anomaly scoring is a broad research area [35]. Here, the relevant literature is grouped around spacecraft telemetry AD and the type of anomaly evidence each family produces [35, 6, 21].

The main anomaly-scoring families carried forward in this thesis are summarized in Table 2.3. The methods are grouped by the kind of anomaly evidence they produce, because that evidence determines whether they can support causal event formation, compact explanation, and MCU-class implementation.

Table 2.3: Anomaly-scoring families considered in this thesis, organized by how anomaly evidence is produced.

ID	Family	Mechanism	Score	Core assumption
F1	Rules, limits, simple statistics	Define a nominal envelope using fixed or adaptive limits, optionally mode-conditioned.	Degree of violation beyond the envelope, per sample or per window.	Nominal behaviour stays within a meaningful envelope; violations indicate abnormality.
F2	Forecasting / prediction residual	Predict expected telemetry from history and/or context.	Residual between observed and predicted values, optionally aggregated over time.	Nominal dynamics are predictable; anomalies produce larger residuals.
F3	Reconstruction	Learn a compact nominal representation and reconstruct the input.	Reconstruction error between the original and reconstructed sample or window.	Nominal telemetry lies near a learnable low-dimensional structure; anomalies reconstruct poorly.
F4	Proximity / distance	Compare each instance with nominal references, neighbours, or prototypes.	Distance to nearest nominal examples or representative prototypes.	Nominal instances lie near similar nominal references; anomalies are farther away.
F5	Distribution / density	Model the nominal probability distribution or density.	Improbability score, such as negative log likelihood, tail probability, or low density.	Nominal instances occur in high-probability regions; anomalies occur in low-probability regions.
F6	Boundary / ensembles	Learn a normal region or combine multiple weak outlier detectors.	Degree of violation relative to a learned boundary, or aggregated ensemble outlier score.	Nominal telemetry occupies a learnable region; anomalies fall outside it or are repeatedly isolated.

For on-board event alerting, the algorithm name is less informative than the evidence representation. A suitable method should produce evidence that is interpretable, causal, robust to telemetry artifacts, and cheap enough for MCU-class execution.

These families are established anomaly-scoring approaches. The taxonomy itself is not a thesis contribution. Its role in this chapter is to define the candidate evidence mechanisms that later screening can assess against the Delfi Twin telemetry, explainability, label, event-level, and embedded constraints.

2.8.2. Expected Thermal Behaviour and Prediction Models

Prediction-based AD is particularly relevant for temperature telemetry because spacecraft thermal signals are shaped by recurring orbital and environmental context. In a residual-based detector, this context is used to estimate the nominal temperature, and anomaly evidence is derived from the difference between the observed and predicted values. The quality of this expected-temperature model therefore directly affects the reliability of the residuals used for detection. However, modelling spacecraft telemetry is challenging because thermal behaviour can depend on multiple coupled variables, delayed responses, and changing mission conditions. Wang et al. similarly note that the "high dimensionality, strong inter-variable coupling, and long-range dependencies in telemetry data...make straightforward modelling approaches inadequate for capturing the underlying patterns" [71].

Spacecraft telemetry prediction techniques for time-series data are typically grouped into 3 categories: physical modelling approaches, data-driven models, and hybrid approaches [71, 42]. Physical modelling methods construct mathematical representations of the telemetry. Data-driven models encompass statistical and Machine Learning (ML) approaches. Hybrid approaches use a mixture of physics-based and data-driven models. For this thesis, it is useful to organize them by how they produce an expected nominal temperature for residual generation. Table 2.4 summarizes their prediction mechanisms.

Table 2.4: Candidate predictor families and preliminary family-level screening for the F2 residual-generation block.

ID	Family	Prediction mechanism	Screening rationale
P1	Parametric / physics-shaped	Predicts temperature using a compact functional form designed to capture orbit-scale thermal structure (e.g., heating/cooling regimes, seasonal offsets, or low-order harmonics).	Simple, interpretable, and physically plausible, while preserving orbital structure with minimal implementation complexity.
P2	First-principles / physical state	Predicts temperature by explicitly evolving thermal states using an energy-balance equation, RC thermal network, or reduced-order state-space model.	Requires orbital, attitude, geometry, and material parameters that are often unavailable or difficult to maintain operationally [74].
P3	Statistical / additive time-series	Uses structured temporal models such as autoregression, trend-seasonal decomposition decomposition, harmonic regression, or additive smoothers.	Lightweight and interpretable, but limited by linearity or stationarity assumptions, which may reduce accuracy for spacecraft thermal data.
P4	Data-driven ML	Learns nonlinear mappings from telemetry history (and optional context) using methods such as tree ensembles, neural sequence models, or transformers.	Can be powerful, but impose higher implementation complexity, reduced interpretability, and significant embedded deployment overhead [71].
P5	Hybrid / physics-informed	Combines physical or parametric models with telemetry-driven residual correction or soft constraints.	Retained when hybrid corrections remain bounded, lightweight, and interpretable; preserves physical structure while improving adaptability [42].

In the later predictor-selection chapter, these families are not ranked as generally better or worse approaches. They are screened for a specific deployment role: generating residuals that support event detection on MCU-class hardware. The selected predictor must capture the dominant nominal thermal behaviour without smoothing away residual structure that is needed to detect anomalies.

For example, Autoregressive model with eXogenous inputs (ARX)/Autoregressive Moving Average with eXogenous inputs (ARMAX)-style models are lightweight statistical comparators that predict future values from past samples and optional exogenous context. Their usefulness for spacecraft thermal telemetry depends on whether the selected history length and context variables can capture non-stationary orbital and seasonal structure. Similarly, adaptive correction can reduce residual bias when a fixed predictor becomes stale, but unconstrained adaptation may update the model toward abnormal behaviour and weaken the residual signal needed for detection.

The predictor literature therefore raises a practical trade-off. The predictor should be accurate enough to produce useful residuals, but not so adaptive or complex that it absorbs abnormal behaviour, becomes difficult to explain, or exceeds embedded constraints. For thermal AD, nominal temperature changes are often large and structured by orbital illumination, seasonal context, attitude, and spacecraft operating mode. Chapter 9 carries forward a small set of candidates from these families and performs the thesis-specific predictor down-selection.

2.8.3. Implications for this Thesis

Algorithm names alone do not determine deployment suitability. For on-board spacecraft telemetry AD, the relevant questions are how anomaly evidence is produced, what assumptions the method makes about telemetry coverage and stationarity, whether it can operate causally, whether it requires labelled faults, whether it can run within embedded constraints, and whether its evidence can be explained in measurement terms.

This point is central for residual-based methods. Residual evidence is attractive because it remains physically interpretable: an observed temperature is hotter or colder than expected by a measurable amount. However, residual methods also introduce predictor-validity risk. If the expected-behaviour model becomes stale or misses the current operating context, residual evidence may reflect model mismatch rather than abnormal spacecraft behaviour. The literature therefore motivates screening complete pipelines, not only scoring algorithms.

2.9. Score-to-Event Decision Logic and Explainable Alerting

The method families reviewed in Section 2.8 define how telemetry can be converted into anomaly evidence. Deployable spacecraft AD requires an additional step: anomaly evidence must be converted into bounded, interpretable, and downlink-compatible alert events. Point-wise anomaly evidence is not yet an operational product. Whether derived from residuals, reconstruction errors, distances from nominal examples, or likelihood estimates, it must be stabilized over time, bounded as an event, summarized, and communicated in a form that supports ground-side interpretation.

This section reviews the decision and explainability concepts needed for that conversion. Subsection 2.9.1 explains why anomaly scores should not be treated directly as operational alerts. Subsection 2.9.2 reviews instantaneous thresholding and cumulative-evidence methods as complementary mechanisms for abrupt and persistent deviations. Subsection 2.9.3 reviews lifecycle concepts such as persistence, hysteresis, minimum duration, and gap handling. Subsection 2.9.4 reviews detector-faithful explainability as a suitable alerting scope for constrained on-board systems.

2.9.1. Why Anomaly Scores Are Not Operational Alerts

After scoring, the detector must decide whether the evidence represents an event worth reporting. For on-board use, this step requires more than thresholding: the logic must suppress short nuisance excursions, preserve event timing, handle gaps, and record why the event was declared or cleared. The score-to-event stage therefore defines the operational alert product, not merely the numerical anomaly score.

A conventional alerting mechanism already exists in spacecraft telemetry monitoring: observed telemetry values are compared against predefined upper and lower limits. Herrmann et al. note that spacecraft telemetry is typically analysed in the time domain using simple limit checking, while also emphasizing that fixed thresholds can be limiting for dynamic systems and have motivated adaptive limit-checking approaches [27]. This threshold-based philosophy is also present in the Delfi-PQ heritage design. Radu et al. describe the Delfi-PQ Fault Detection, Isolation and Recovery (FDIR) system as simple and deterministic, based on threshold exceedance for components and subsystems, with thresholds allowed to depend on technical specifications and the current operational mode [54]. Upper/lower limit checking is therefore an appropriate heritage baseline for this thesis, rather than an arbitrary comparator.

However, limit violations do not by themselves define operator-usable alert products. In many AD pipelines, the score-to-alarm stage is described simply as *thresholding*, since a common baseline is to declare an anomaly when a score exceeds a limit, $s(t) > \tau_s$ [9]. Yet a pointwise threshold crossing may flicker because of noise, a single-sample excursion may be a benign spike, a sustained small residual shift may never exceed a high instantaneous threshold, and a telemetry gap may interrupt an event without proving recovery. Threshold crossings are treated in this thesis as evidence for event formation, not as operational alerts by themselves.

2.9.2. Thresholding and Cumulative Evidence

It is useful to distinguish three thresholding levels. The first is direct upper/lower limit checking on the raw telemetry value, such as a maximum or minimum allowable temperature. The second is pointwise thresholding of a derived anomaly score, such as a prediction residual, reconstruction error, or normalized residual. The third is event-level threshold logic, where threshold crossings are treated as evidence that must persist, accumulate, clear, or terminate according to lifecycle rules. This thesis uses the second and third levels: raw temperature limits remain necessary for deterministic safety monitoring, but the residual-to-alert layer asks whether the temperature is abnormal relative to expected thermal behaviour.

The decision logic is structured using concepts from Statistical Process Control (SPC), where monitored variables are compared against expected behaviour using control charts [49]. When applied to residual scoring, the monitored signal is the residual-derived evidence stream, such as $r_{j,t}$, $|r_{j,t}|$, or a normalized residual $z_{j,t}$. Control-chart logic is commonly divided into memory-free and memory-type approaches.

Memory-free thresholding provides fast response to large residual excursions. A Shewhart-like rule compares the current residual evidence directly with a threshold, analogous to checking whether a monitored statistic exceeds a control limit. In essence, this functions as pointwise thresholding: the limits are often set at $\pm 3\sigma_{r,j}$ about the nominal value, and any score that violates this range is deemed anomalous [36]. This approach is simple, interpretable, and computationally cheap, making it attractive for embedded use. Its weakness is that it is less sensitive to small or moderate shifts that remain below the instantaneous threshold.

Memory-type detection accumulates weak evidence over time. Cumulative Sum (CUSUM) and Exponentially Weighted Moving Average (EWMA) are common examples of memory-type control charts [49]. These methods are better suited to subtle but persistent changes because they use both current and past evidence. CUSUM-style approaches are well suited to thermal telemetry because they accumulate directional evidence over time, making them suitable candidates for detecting sustained positive (hotter-than-expected) or negative (cooler-than-expected) deviations. EWMA-style methods are also useful as smoothing or memory-type monitoring approaches. The relative suitability of these mechanisms depends on the scoring representation, alerting objective, and deployment constraints.

Instantaneous and cumulative evidence therefore serve complementary roles. Instantaneous evidence supports fast detection of large abrupt deviations, while cumulative evidence supports weaker persistent shifts.

2.9.3. Event Lifecycle Concepts

Practical alert generation often requires further judgment, because threshold crossings are not yet operational alerts. Real spacecraft telemetry can contain isolated spikes, near-threshold noise, missing data, glitches, and predictor mismatch, any of which can produce misleading short-lived evidence. Alarm-management practice therefore uses mechanisms such as deadbands and time-delay logic to reduce nuisance alarms [43]. Typical mechanisms include persistence requirements (do not alarm too quickly), hysteresis logic (do not clear too quickly), minimum-duration constraints, merge/split handling for nearby violations, gap reset under missing telemetry, and severity assignment [20, 43, 61]. These rules are not minor implementation details; they determine whether an alert is trustworthy enough to warrant operator attention.

Without lifecycle logic, a detector may produce outputs that are operationally misleading: too many short events, unrelated deviations merged into one event, events extended through unobserved intervals, or events cleared before the underlying evidence has stabilized. Lifecycle design is therefore the step that converts residual evidence into bounded spacecraft events that operators can interpret and trust.

2.9.4. Explainability for Operational Alerting

Explainability is a core requirement because the alert must help operators understand why telemetry was considered abnormal. Li et al. [37] identify several relevant properties for explainable AD, including detection fidelity, comprehensibility, generality, scalability, and complexity. Fidelity, the "explanation faithfulness", defines how closely the provided explanations reflect the detector's true scoring and decision mechanism. For this thesis, the most relevant form of explainability is detector-faithful explanation. This is not post-hoc explainability in the sense of explaining an opaque model after inference. Instead, the alert reports quantities already used by the detector, such as residual magnitude, direction, persistence, trigger condition, and event timing.

This is different from root-cause diagnosis. A detector-faithful alert can say that the battery temperature was hotter than predicted, that the event began because of cumulative positive residual evidence, and that the event cleared by hysteresis. It cannot, by itself, prove whether the cause was a heater issue, self-heating, sensor drift, or battery degradation. The alert payload should therefore be compact, evidence-based, and honest about its scope: it should report detector evidence directly rather than rely on a separate post-hoc explanation model.

The decision and explainability literature reinforces the pipeline view introduced in Section 2.2. For on-board use, anomaly evidence must be converted into lifecycle events with supporting detector evidence and compact alert content. This motivates the residual-to-alert decision layer selected in Chapter 8 and defined in Chapter 10.

2.10. General Literature Gap and Implications

The reviewed literature establishes the main components needed for deployable spacecraft AD. However, it also shows that on-board thermal anomaly alerting requires these components to be integrated into a complete deployment pathway for MCU-class spacecraft hardware.

FM literature defines the operational chain and shows why detection must be distinguished from diagnosis, decision, response, and adaptation. Spacecraft telemetry AD literature provides scoring and prediction families. Embedded-systems literature identifies compute, memory, power, bus, and packet constraints. Dataset literature shows why public spacecraft telemetry is valuable but often insufficient for mission-specific fault labels. Decision-logic literature shows that anomaly evidence must be converted into bounded events. Explainability literature motivates alerts that report detector evidence rather than unexplained anomaly flags.

The general gap is therefore not the absence of AD algorithms. The gap is the lack of an integrated workflow that connects operational role, telemetry observability, anomaly semantics, ground-truth strategy, score-to-event logic, alert content, embedded feasibility, and claim boundaries. For downlink-constrained spacecraft, this integration matters because telemetry access may be intermittent, on-board resources may be limited, and operators need compact evidence rather than continuous score streams.

The following chapters in Part I apply this general gap to the Delfi Twin case study. Chapter 3 derives the mission-specific operational and embedded requirements. Chapter 4 derives the temperature-telemetry and anomaly scope. Chapter 5 then synthesizes the Part I foundation into the research objective, research questions, thesis contributions, and research plan.

3

Mission Use Case and Requirements

Chapter 2 reviewed the broader Fault Management (FM), spacecraft telemetry Anomaly Detection (AD), embedded deployment, score-to-event, and explainable-alerting literature. This chapter applies that foundation to Delfi Twin as a representative downlink-constrained small-spacecraft case study. Its purpose is to turn the literature-level deployment problem into concrete operational, timing, responsibility, and embedded-resource requirements for the remainder of the thesis.

The central operational motivation is Time to Awareness (TTA). Telemetry may be sampled on board long before it is reviewed on the ground, because communication opportunities are intermittent and full-resolution telemetry downlink is bandwidth-limited. A ground-only monitoring approach can therefore delay awareness of abnormal behaviour. On-board AD is introduced here as an advisory function that detects suspected events, packages compact evidence, and supports later prioritization of relevant context windows for downlink.

This chapter defines the requirements that later design choices must satisfy. Section 3.1 defines the operational use case and scope. Section 3.2 applies the FM concepts from Section 2.1 to allocate responsibilities between the spacecraft and ground operators. Section 3.3 derives the real-time alerting requirement and latency budget. Section 3.4 translates embedded deployment constraints into resource-budget assumptions. Section 3.5 summarizes the requirements passed forward to the telemetry-scope, dataset, pipeline-selection, embedded-implementation, and evaluation chapters.

3.1. Operational Use Case and Scope

The on-board AD function is scoped as an advisory early-awareness function for ground-operator prioritization and initial assessment. It does not replace the existing FM process, perform autonomous recovery, or provide definitive root-cause diagnosis. Its role is to identify temperature behaviour that deviates from expectation, form event records, and report compact evidence that helps operators decide what requires further attention.

This scope follows from the FM concepts reviewed in Section 2.1. In this thesis, the spacecraft-side function performs detection, event formation, and alert packaging. Diagnosis, response selection, and model or procedure updates remain ground-side responsibilities. The resulting operational use case is summarized in Table 3.1.

The alert content is therefore treated as evidence for operator assessment rather than as an on-board diagnosis. Root-cause interpretation requires additional telemetry, mission context, and operational judgement that remain ground-side responsibilities.

The use case also assumes that alerts may need to wait for a later downlink opportunity. Since contact opportunities are intermittent, the detector cannot assume immediate transmission. The alert queue therefore acts as a compact event buffer: it stores event summaries rather than continuous high-rate telemetry.

Table 3.1: Operational use case and scope boundaries for on-board AD on Delfi Twin.

Requirement area	Scope in this thesis
Telemetry source	Temperature housekeeping telemetry from selected Delfi Twin nodes.
Operating mode	`Nominal Mode` only. Mode-aware operation is left as future work.
On-board role	Detect suspected events, form bounded events, and package compact alerts.
Ground role	Perform detailed diagnosis, decide response actions, and manage model or configuration updates.
Downlink role	Prioritize compact alerts and relevant context windows instead of continuous full-resolution telemetry.
Timing role	Prepare alerts sufficiently fast to transmit in the next available downlink frame.

If each alert has bounded size B_{alert} and at most N_{queue} alerts are buffered over a blackout interval, the required alert-buffer capacity is

$$B_{\text{queue}} = N_{\text{queue}} B_{\text{alert}} \tag{3.1}$$

This queued volume should remain small enough to fit within the next planned downlink allocation. Alert size is therefore treated as a deployment constraint alongside compute, memory, bus use, and power.

The thesis scope is restricted to *Nominal Mode* operation. This prevents expected thermal changes from mode transitions, special operations, or unusual spacecraft configurations from being treated as anomaly evidence. Mode-aware detection is required for operational deployment, but it is left as a future extension rather than included in the first prototype requirement set.

The detector operates on time-ordered housekeeping samples using an on-board Mission Elapsed Time (MET) or equivalent monotonic timebase. Converting event times to absolute UTC and matching them with ground-received telemetry remain ground-side tasks. For causal on-board detection, sample order, elapsed time, and approximate cadence are sufficient.

3.2. Fault-Management Responsibility Allocation

The responsibility split in Figure 3.1 makes the advisory scope explicit. The spacecraft-side function detects unexpected thermal behaviour, forms event records, and packages the evidence needed for ground review. It does not act autonomously on detected anomalies.

This allocation reduces delay between event occurrence and operator awareness while avoiding over-claims about on-board diagnosis or recovery. Ground operators remain responsible for interpreting the alert in mission context, comparing it with other telemetry, selecting any response, and managing vetted updates to thresholds, models, or procedures.

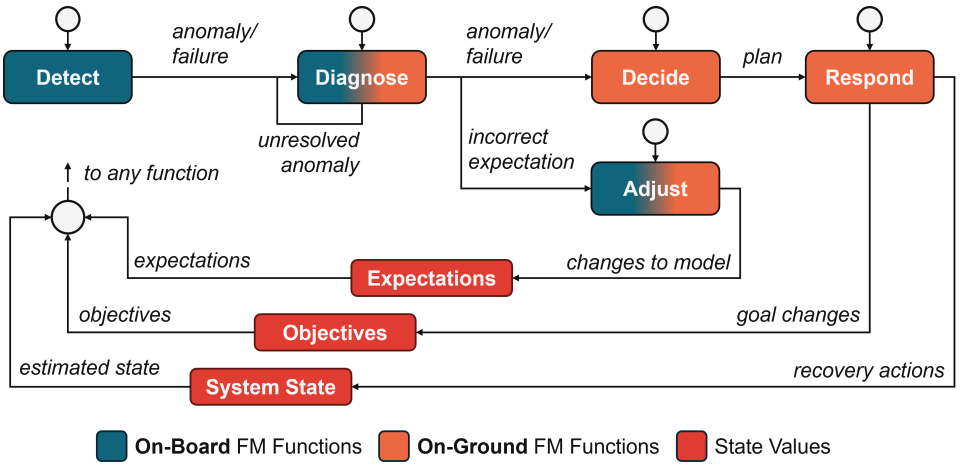


Figure 3.1: Operational FM function flow and responsibility split for Delfi Twin (adapted from [20])

In this allocation, the on-board function is responsible for detecting unexpected telemetry behaviour, identifying the affected temperature channel, summarizing event timing, direction, and magnitude, reporting the detector evidence behind the declaration, and preserving whether the event cleared normally or was interrupted by telemetry loss.

The ground segment remains responsible for diagnosis, operational classification, response selection, model or procedure updates, and decisions about whether additional telemetry should be requested or prioritized. This allocation keeps the on-board function advisory: it reduces delay to operator awareness while leaving diagnosis, response selection, and model management to ground operations.

3.3. Real-Time Requirements and Latency Budget

FM literature motivates timing analysis through Time to Criticality (TTC). In a complete FM design, detection, diagnosis, decision, and response should occur before a fault propagates to an unacceptable consequence. Estimating TTC for all relevant Delfi Twin thermal fault paths would require a full Failure Modes, Effects, and Criticality Analysis (FMECA) and Fault tree analysis (FTA) [20], which is outside the scope of this thesis.

This chapter therefore defines a narrower on-board timing requirement. The AD function shall process native housekeeping telemetry at the assumed 15 s cadence and prepare a detect-and-notify alert within a 60 s alerting frame. This target does not imply that all thermal faults have a 60 s TTC; it defines the prototype timing requirement for detection and alert packaging.

Operational alerting delay includes more than detector computation. The on-board latency terms are listed in Table 3.2, and the corresponding latency budget is defined as:

$$t_{\text{onboard}} = t_{\text{sensor}} + t_{\text{bus}} + t_{\text{preproc}} + t_{\text{persist}} + t_{\text{compute}} + t_{\text{pack}} \leq 60\,\text{s} \tag{3.2}$$

Persistence is the main latency term controlled by decision-logic design. A single sample may be insufficient to distinguish a true event from an outlier, so the detector may require repeated evidence before declaring an event [20]. If the event logic requires N (near-)consecutive samples that satisfy the start condition, and samples arrive every t_s, the confirmation delay is

$$t_{\text{persist}} = (N-1)t_s. \tag{3.3}$$

For $t_s = 15\,\text{s}$, each additional confirmation sample adds one cadence interval before declaration. Persistence can suppress isolated noise or spikes, but excessive persistence delays awareness. The decision logic therefore treats persistence as a controlled part of the 60 s alerting budget.

The full operational response time also includes communication and ground-operations delays. A complete response must satisfy

$$t_{\text{end-to-end}} = t_{\text{onboard}} + t_{\text{contact\_wait}} + t_{\text{gnd\_ingest}} + t_{\text{ops\_decide}} + t_{\text{uplink}} + t_{\text{execute}} < \text{TTC} \tag{3.4}$$

This thesis evaluates only the on-board portion of that chain. The detector must process samples causally, without future information, and its worst observed processing time should remain small relative to the 15 s input cadence and the 60 s alerting frame.

The remaining terms still shape the operational motivation. Contact waiting time, downlink priority, ground ingestion, operator review, uplink delay, and response execution all affect final operator TTA. They are outside the embedded timing measurement, but they motivate compact on-board alerts that can be queued and prioritized for downlink.

The resulting on-board real-time AD requirements are defined in Table 3.3.

3.4. Embedded Platform and Resource Budget Assumptions

Section 2.4 reviewed the embedded constraints relevant to on-board AD: compute time, Random-Access Memory (RAM), flash footprint, power consumption, bus load, packet size, and memory technology. This section translates those constraints into Delfi Twin resource-budget assumptions and prototype acceptance targets. These targets are later used to assess timing, memory footprint, alert-packet size, and resource use in the embedded implementation and evaluation chapters.

Table 3.2: End-to-end AD latency terms, grouped into on-board and remaining operational terms

Latency term	Symbol	Definition
	On-Board Latencies	
Sensor latency	t_{sensor}	Time from physical temperature change to sampled sensor output, including sensor response, analog-to-digital conversion, and driver delay
Bus transfer	t_{bus}	Time to transfer the sampled reading into the On-Board Computer (OBC) telemetry buffer, including bus transaction and brief driver or Interrupt Service Routine service.
Preprocessing	t_{preproc}	Delay from telemetry conditioning before detection, such as filtering, smoothing, resampling, or time alignment.
Persistence	t_{persist}	Delay between first anomalous evidence and event declaration when repeated evidence is required.
Compute (detect+alert)	t_{compute}	Worst-Case Execution Time (WCET) to update the predictor, residual evidence, event state, and alert fields.
Pack/encode	t_{pack}	Time to serialize the alert fields into a queued packet or frame for downlink.
	Remaining End-to-End Latencies	
Contact wait	$t_{\text{contact\_wait}}$	Time from alert queueing to the next available ground contact; zero if already in contact.
Ground ingest	$t_{\text{gnd\_ingest}}$	Time from ground-station receipt to availability in the operator or ground-automation system.
Operator decision	$t_{\text{ops\_decide}}$	Time for operators or ground automation to interpret the alert and select a response.
Uplink delivery	t_{uplink}	Time to schedule, transmit, and receive the selected command on board.
Execute on board	t_{execute}	Time after command receipt for flight software, subsystem logic, or actuator response to take effect.

Table 3.3: On-board real-time AD requirements

Requirement	Definition
Ingest cadence	Temperature telemetry shall be processed at the native 15 s sampling period.
Frame deadline	Alerts shall be generated within a 60 s downlink frame.
Persistence budget	Persistence settings shall ensure confirmation delay remains within the 60 s latency budget.
Store-and-forward	Alerts shall be queued during communication blackouts and transmitted after contact resumes.
Timebase	A common timebase shall be used for event correlation across resets.
Bandwidth efficiency	Alerts shall be compact and include only relevant context windows rather than full-resolution telemetry.

3.4.1. Target Platform Context

Delfi Twin targets an STM32L4-class Microcontroller Unit (MCU) for the On-Board Computer (OBC). The embedded prototype is implemented on an STM32L476RG development board from the same STM32L4 family, providing a representative platform for MCU-class feasibility assessment. Table 3.4 compares this development board with the Delfi-PQ MSP432 reference board and the STM32L496 target board.

The platform choice tests whether the pipeline can run within a realistic MCU-class resource envelope, rather than whether high-performance inference is possible. This constrains the later design toward causal processing, bounded working state, simple arithmetic, and compact memory use. Methods requiring long input histories, large stored models, expensive matrix operations, or high-rate feature extraction are less suitable unless simplified for embedded execution.

Table 3.4: Processing and resource comparison of Delfi-PQ and Delfi Twin boards.

Board	Flash [kB]	RAM [kB]	Power [mW]	Clock [MHz]	Performance
MSP432 [12]	2048	256	23	48	1.22 DMIPS/MHz (Dhrystone 2.1); 163.68 CoreMark® (3.41 Coremark/MHz)
STM32L476RG [13]	1024	128	26	80	1.25 DMIPS/MHz (Dhrystone 2.1); 273.55 CoreMark® (3.42 Coremark/MHz)
STM32L496 [14]	1024	320	33	80	1.25 DMIPS/MHz (Dhrystone 2.1); 273.55 CoreMark® (3.42 Coremark/MHz)

3.4.2. Memory Strategy

The memory-technology considerations reviewed in Section 2.4 motivate a tiered storage strategy for on-board AD. As summarized in Table 3.5, on-chip flash is reserved for firmware and static constants, while on-chip Static Random-Access Memory (SRAM) holds time-critical working data during detector execution. Ferroelectric Random-Access Memory (FeRAM) is suitable for small frequently updated persistent state, and external flash is better suited to larger sequential records such as logs or stored event snapshots.

This strategy keeps active detector state in SRAM, avoids frequent small writes to flash, and assigns persistent records to memory technologies that match their update pattern. A flight implementation with stored event snapshots or context windows would require an appropriate non-volatile storage region, such as external flash.

Table 3.5: Memory map for on-board AD: memory characteristics and intended roles.

Memory region	Key characteristics (why it matters)	Primary role in AD
On-chip flash	Non-volatile. Finite P/E endurance. Erase occurs at page granularity, so frequent small updates concentrate wear.	Firmware, AD code, and static constants.
On-chip SRAM	Volatile. Fast random access. Contents are lost on reset or power loss.	Working memory during AD execution.
External FeRAM	Non-volatile. Very high write endurance. Fine-grained updates (well-suited to frequently rewritten small data).	Small, frequently updated persistent state.
External flash	Non-volatile bulk storage. Erase occurs in large blocks; best used with sequential or append writes.	Bulk records written sequentially (logs and event snapshots).

3.4.3. Prototype Resource Targets

The resource targets define what it means for a candidate AD pipeline to remain plausible for MCU-class deployment. They cover processing time per sample, flash footprint, RAM use, alert-packet size, bus load, and power consumption. At prototype stage, these targets are derived from total device resources and reduced using design margins. A flight implementation should recompute the caps once the resource allocation available to AD is known.

Resource margins depend on both mission maturity and verification method, such as estimation, analysis, or measurement [57]. In this thesis, the embedded prototype is measured on target hardware but is not a fully integrated flight build. The Critical Design Review (CDR)-level margins in Table 3.6 are therefore used as prototype budget targets for the essential computing resources.

The bus-utilization target is treated more conservatively than the generic data-interface margin. The AD pipeline considered here interacts with MCU-level I2C/SPI sensor buses, where excessive utilization can introduce latency and jitter. As a conservative rule of thumb, Controller Area Network bus (CAN) bus designs are often kept below an average load of approximately 50 % [16], and CAN is also used as a spacecraft data bus [65]. This thesis therefore adopts a 50 % bus-utilization cap to limit AD-induced jitter and protect Worst-Case Execution Time (WCET).

Using these margins and the assumption of full-device resource allocation for prototype budgeting, the resulting resource targets are given in Table 3.7.

Table 3.6: Flight-software resource margins by development maturity (adapted from [57])

Resource	SRR [%] (Estimate)	PDR [%] (Analysis)	CDR [%] (Analysis/Measured)	Flight [%] (Measured)
Average CPU usage	50	50	40	30
Deadlines (slack)	50	30	20	10
Non-writeable NVM	50	30	20	0
Writeable NVM (flash)	50	50	40	30
RAM	50	50	40	30
Data interfaces (avg load)	40	30	20	10

SRR: System Requirements Review, PDR = Preliminary Design Review, CDR = Critical Design Review

Table 3.7: Prototype resource metrics, margins, and baseline budgets

Metric	Margin [%]	Budget	Implication
RAM (Peak)	40	L476: $\leq$ 80 kB of 128 kB L496: $\leq$ 192 kB of 320 kB	Confirms feasibility on the development board and leaves room for model changes and mode-aware extensions.
Flash (Image)	40	≤ 600 kB of 1024 kB	Avoids use of external memory for code, preserves timing and energy behaviour, and allows for future code growth.
CPU usage	40	$\leq 60\,\%$ average over the AD cycle	Keeps compute load within the allocated budget and protects against timing jitter.
Bus load	50	$\leq 50\,\%$ average load on shared bus reads	Limits bus-induced latency and protects WCET.
Power (MCU)	–	≤ 50 mW worst case; typical 26 mW to 33 mW	Small relative to the approximately 2 W orbit-average spacecraft power budget.

The targets are deliberately conservative. The detector should use only a small fraction of the sample interval because the OBC must also support other spacecraft functions. RAM and flash usage should remain low enough to coexist with flight software, alert packets should fit intermittent downlink constraints, and bus and power overhead should not become dominant spacecraft loads.

These targets are evaluated in Chapters 11 and 12. The measured resource scale is also contextualized against reported embedded AD implementations introduced in Section 2.4.

3.5. Chapter Summary: Requirements Passed Forward

This chapter derived the operational and embedded requirements that bound the remainder of the thesis. The on-board function is scoped as an advisory early-awareness system: it detects suspected temperature events, forms bounded event intervals, and packages compact evidence-based alerts. It does not perform autonomous recovery or definitive root-cause diagnosis.

The chapter also defined timing and resource expectations. The AD function shall process 15 s telemetry samples, prepare alerts within a 60 s window, and remain feasible on MCU-class hardware. The embedded design is constrained by compute, memory, packet size, bus, and power limits, with CDR-level margins used as prototype acceptance targets. The resulting requirements are summarized in Table 3.8.

Table 3.8: Derived deployment requirements and design implications carried forward from this chapter.

Derived requirement	Design implication
Advisory on-board detection only	Pipeline selection, alert-payload design, operational value assessment
Ground retains diagnosis, decision, response, and adaptation	Explainability scope, alert claim boundaries, deployment-readiness assessment
Nominal Mode only	Telemetry scope, clean benchmark construction, limitations
15 s ingest cadence	Decision timing, embedded replay, processing-time verification
60 s alert-frame target	Persistence choice, time-to-awareness interpretation
Compact queued alert packets	Alert-payload fields, packet encoding, downlink relevance
MCU-class compute and memory constraints	Pipeline screening, embedded implementation, resource verification
Limited power and bus impact	Architecture screening, deployment-readiness discussion
Causal operation without future samples	Predictor, decision logic, embedded implementation

Together, these requirements constrain the broader deployment problem for the Delfi Twin case study by defining the operational role, timing target, responsibility split, and embedded resource assumptions. Chapter 4 completes the scope definition by identifying the temperature telemetry channels, nominal thermal behaviours, anomaly vocabulary, telemetry-pathology categories, and thermal fault families considered in the thesis. Chapter 5 then synthesizes Chapters 2 to 4 into the formal research questions and research plan.

4

Telemetry Observability, Anomaly Semantics, and Fault Classes

Chapter 2 reviewed spacecraft telemetry Anomaly Detection (AD), telemetry realism, dataset limitations, score-to-event methods, embedded deployment, and explainable alerting. Chapter 3 then translated that foundation into the Delfi Twin mission use case, deriving the operational role, alerting requirement, responsibility split, and embedded resource assumptions used in the remainder of the thesis. This chapter completes the Part I scope foundation by defining what the detector can observe and what kinds of thermal behaviour it is expected to distinguish.

The starting point is telemetry observability rather than model selection. A detector cannot provide early awareness if the relevant physical behaviour has no observable precursor, is not sensed by the available telemetry, or is not represented across the operating regimes to be monitored [6]. The chapter therefore assesses why temperature telemetry is a suitable first target for the prototype, which nominal behaviours should be modelled or tolerated, and which telemetry-quality and thermal-fault behaviours are carried forward into the benchmark and evaluation.

The chapter proceeds in five steps. It first identifies the available Delfi Twin telemetry and motivates Electrical Power System (EPS) temperature telemetry as the initial scope. It then defines nominal thermal behaviour, introduces the anomaly-pattern vocabulary used for temperature deviations, separates telemetry pathologies from physical thermal faults, and states the scope boundaries that govern later claims.

4.1. Delfi Twin Telemetry Characteristics

Delfi Twin telemetry is expected to closely follow the Delfi-PQ housekeeping structure. The available fields include measurements from the EPS, On-Board Computer (OBC), Attitude Determination and Control System (ADCS), Communications (COMMS), On-Board Data Processing (OBDP), and related support boards. These fields include numeric quantities such as voltage V, current I, and temperature T, together with status flags, enumerations, and mode indicators that provide operational context.

This thesis uses EPS telemetry as the starting point for the on-board AD prototype because it is observable, interpretable, and operationally significant. Voltage, current, and temperature channels can reveal undervoltage, over-current, load changes, panel degradation, and thermal issues, while status fields help interpret the operating context. EPS anomalies also have direct mission impact because they can affect the power budget, trigger resets, or require throttling. This selection follows the telemetry suitability criteria discussed by Bieber et al. [6], and the EPS is a well-studied target for telemetry-based AD [66, 31, 52].

Table 4.1 summarizes the available EPS housekeeping parameters by component family and signal type. The inventory identifies the voltage, current, temperature, status, and auxiliary housekeeping fields available to the case study. From these signals, selected EPS temperature channels are carried forward as the primary telemetry source for benchmark construction and detector evaluation.

The on-board detector is assumed to receive conditioned housekeeping telemetry rather than raw packet fields. Raw values must first be decoded, calibrated to engineering units, rejected or flagged when outside valid ranges, and aligned to a common time base where required. Relevant status and mode flags should also be aligned where available. The scope defined here therefore begins after basic telemetry conditioning; full raw-packet decoding and flight-software integration are addressed only as future deployment tasks.

Table 4.1: EPS telemetry by component family: instances and per-instance signal types (counts). Totals reflect unique payload parameters (wrappers de-duplicated).

Component family	Instances	Voltage /inst	Current /inst	Temp /inst	Status /inst	Other /inst	Total
Battery (pack)	1	2	1	2	2	1	8
Power buses (Bus 1..4)	4	1	1	0	1	2	20
Solar panels (±X, ±Y)	4	1	1	1	2	0	20
MPPT channels (±X, ±Y)	4	1	1	0	1	0	12
Panel cells (±X, ±Y)	4	1	1	0	1	0	12
Power-rail monitors (INA: Internal, Unregulated)	2	1	1	0	1	0	6
Total		20	19	6	24	9	78

4.2. Literature-Derived Rationale for Temperature Telemetry

Temperature telemetry is selected because it is physically meaningful, widely available, and compatible with lightweight expected-behaviour modelling. It is coupled to component health, battery safety, subsystem self-heating, heater behaviour, illumination, thermal interfaces, and surface-environment interactions. It also tends to vary more slowly than many electrical quantities, which supports causal residual modelling and event-level AD under Microcontroller Unit (MCU)-class constraints.

Voltage, current, status, and mode fields remain essential for diagnosis and future context-aware detection. In this thesis they are treated as supporting context rather than primary detector inputs. Temperature is used as the first target because it provides a practical signal for testing whether an on-board thermal anomaly-awareness function can be made interpretable and resource-feasible.

The selection is justified by four properties:

- **Direct coupling to dominant failure mechanisms**
  Temperature is directly linked to reliability-critical failure mechanisms. Many electronic and battery degradation processes are accelerated by high temperatures, large thermal gradients, or repeated cycling. Temperature telemetry therefore provides a physically meaningful signal for early fault awareness. [32]
- **Board-level coverage with minimal sensing**
  Temperature sensors can provide board-level observability without additional hardware. Heat from a faulty or overactive component can conduct through the board and surrounding structure, making it possible for nearby temperature sensors to detect abnormal behaviour even when the failing component is not directly being measured. Achieving comparable coverage with current or voltage measurements requires additional instrumentation. [29]

- **Slowly varying dynamics well-suited to lightweight models**
  Temperature varies more slowly and smoothly than many electrical signals. Voltage and current can change abruptly with mode transitions, switching events, and load changes, while thermal signals are filtered by thermal inertia. This slow behaviour is advantageous for on-board AD, as simple, lightweight algorithms can build stable baselines and detect gradual shifts that are easier to model robustly with low computational cost. Furthermore, repeatable orbital thermal cycles of periodic sunlight/eclipse transitions and repeated operational modes define an expected pattern against which anomalies can be detected. [41, 29]
- **Ubiquity of sensors and negligible implementation overhead**
  Temperature sensors are already common in spacecraft housekeeping telemetry and typically span numerous subsystems. A temperature-based detector can therefore be implemented without adding new sensing hardware, which is important for small satellites with tight mass, power, and integration constraints. [32]

For these reasons, temperature is adopted as the primary telemetry medium for the first detector scope considered in this thesis. Voltage, current, and status fields remain valuable future extensions, especially for improving context awareness and distinguishing thermal faults from expected operational changes. The available temperature sensors in the Delfi Twin telemetry are summarized in Table 4.2.

Table 4.2: Temperature telemetry channels available for Delfi-PQ/Delfi Twin, with calibrated validity bounds and operating limits.

Channel (location)	Parameter name	Unit	Valid range		Operating limits	
			min	max	min	max
Battery (GG sensor)	`BatteryGGTemperature`	°C	−60	130	−50	70
Battery (TMP20 sensor)	`BatteryTMP20Temperature`	°C	−60	130	−50	70
Solar panel $-X$	`PanelXmTemperature`	°C	−60	130	−60	80
Solar panel $+X$	`PanelXpTemperature`	°C	−60	130	−60	80
Solar panel $-Y$	`PanelYmTemperature`	°C	−60	130	−60	80
Solar panel $+Y$	`PanelYpTemperature`	°C	−60	130	−60	80
MCU/OBC	`MCUTemp`	°C	*No bounds in extracted metadata*			

4.3. Role and Limitations of Historical Delfi Telemetry

The telemetry inventory and temperature-channel rationale describe fields expected to be available to an on-board detector. The historical Delfi telemetry available for this thesis has a different role: it appears to represent ground-received samples rather than a continuous on-board time series. It was therefore assessed separately to determine whether it could support detector replay and quantitative validation.

A continuity check showed that the available historical Delfi telemetry is too sparse for quantitative detector validation. The densest sliding 95 min window contained only 12 telemetry rows, with most samples concentrated in a short contact segment, followed by a long gap and a few samples near the end of the window. Fixed 95 min orbit bins showed the same pattern, with the best bins containing approximately 10 samples. Additional cleaning did not resolve this limitation, indicating that sampling sparsity, not only invalid temperature values or outlier removal, is the limiting factor.

The detector assumes an approximately regular causal on-board input stream for persistence, cumulative evidence, lifecycle start and clear logic, and Time to Awareness (TTA) estimation. Running it on sparse ground-received samples would mainly measure contact-window structure and missing data rather than detector behaviour. The historical Delfi telemetry is therefore useful for real-telemetry context and aggregate thermal-pattern illustration, but not for recall, precision, false-alert-rate, TTA, or lifecycle-performance claims.

4.4. Nominal Thermal Environment and Temperature Behaviour

Before defining anomalies, it is necessary to specify which behaviours should be considered nominal. For Earth-orbiting small spacecraft such as Delfi Twin, nominal temperature behaviour is shaped mainly by three effects:

1. the orbit-scale sunlight and eclipse cycle,
2. attitude, spin, and nutation effects,
3. slow seasonal variation from beta angle (β) and solar irradiance.

Temperature telemetry is shaped by environmental and operational context. Ordinary orbit-scale heating, spin-imprinted ripple, seasonal illumination variation, annual irradiance change, and low-level nominal noise should be modelled or tolerated rather than interpreted as faults. These behaviours define the nominal thermal structure used later for clean-reference construction and residual scoring.

Spacecraft operating mode also affects temperature. Communication activity, payload operation, power conversion, charging state, and internal electronics load can all change self-heating. As stated in Section 3.1, this thesis is restricted to `Nominal Mode`. Mode-aware thermal modelling is retained as a future extension.

Orbit-Scale Day-Night Cycle

For small satellites in Low Earth Orbit (LEO), the dominant driver of thermal behaviour is the orbit-scale day-night cycle. External panels experience direct solar heating in sunlight and radiate to deep space in eclipse, producing large periodic temperature swings. On Delfi-PQ, the solar panels exhibited 50 K to 60 K temperature swings per orbit [5]. In contrast, the internal battery and MCU experienced smaller, lagged swings of around 35 K and 40 K, respectively. The exact amplitude of each temperature node depends on attitude, β, and detailed thermal properties. For AD, the relevant feature is the signal structure: the orbital waveform is large, periodic, repeatable over many orbits, and dependent on operating mode. This orbit-scale behaviour is carried forward as a required nominal component for later clean-reference construction.

Seasonal and Illumination Context

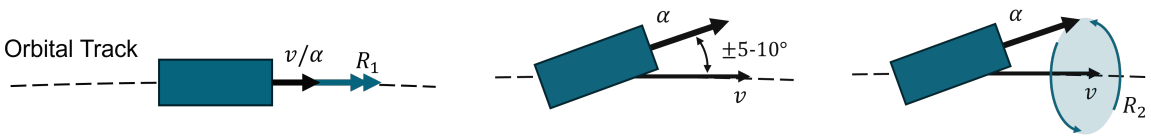
A second nominal driver is the seasonal variation in Sun-spacecraft geometry. Solar beta angle β defines the angle between the Sun vector and the orbital plane. At low β, eclipses are long in each orbit, producing large hot/cold swings and lower average temperatures. At high β, eclipses are short or absent, producing reduced orbit swings and higher average temperatures. This phenomenon is visualized in Figure 4.2. In a Sun-Synchronous Orbit (SSO), the Local Time of the Ascending Node (LTAN) maintains a roughly fixed relationship between the orbital plane and the Sun, but the solar declination varies over the year. As a result, the solar beta angle is not strictly constant. Changes in beta angle affect eclipse duration and the relative timing of heating and cooling within the orbit. In this thesis scope, eclipse geometry and seasonal illumination are therefore treated as nominal context rather than anomaly evidence. [25]

A third nominal driver is annual irradiance variation caused by the eccentricity of Earth's orbit around the Sun. The Earth-Sun distance changes over the year, which produces a slow annual variation in incident solar flux. For Delfi-C3, a clear one-year sinusoidal modulation of internal stack temperature with an amplitude of 3.1 K was observed on the OBC [8]. External panels should expect larger annual changes, due to their direct exposure to the environment and to the β-dependent changes in illumination. In this thesis scope, annual irradiance variation is treated as nominal baseline variation rather than as an anomaly.

Spin and Nutation Imprint

Delfi Twin is a spin-stabilized spacecraft, expected to rotate about its velocity (long) axis at 5 °/s to 10 °/s. In addition, the long axis is expected to nutate within ±5° to 10° of the velocity vector at 1 °/s to 5 °/s. This spin and nutation are visualized in Figure 4.1. They introduce a smaller ripple superimposed on the orbit-scale waveform. The effect is most visible on external panels and more strongly filtered on internal nodes. A comparable imprint can be seen in FUNcube-1 telemetry, shown in Figure 4.3.

In this thesis, spin-imprinted ripple is treated as nominal thermal texture. Later modelling and detection stages should tolerate it unless additional attitude-aware evidence supports a different interpretation.



(a) Spin about the velocity (long) axis

(b) Nutation (precession) cone about the velocity vector

(c) Attitude evolution over one spin cycle

Figure 4.1: Nominal Delfi Twin attitude, showing a 5 °/s to 10 °/s spin about the velocity axis with a ±5° to 10° precession cone, which imprints a small high-frequency ripple on temperatures (adapted from [4]).

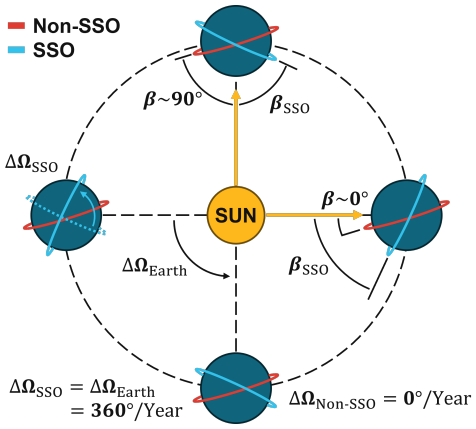


Figure 4.2: Definition of beta angle β and comparison of its large variations in a generic LEO orbit with the bounded, slowly varying β in a sun-synchronous orbit.

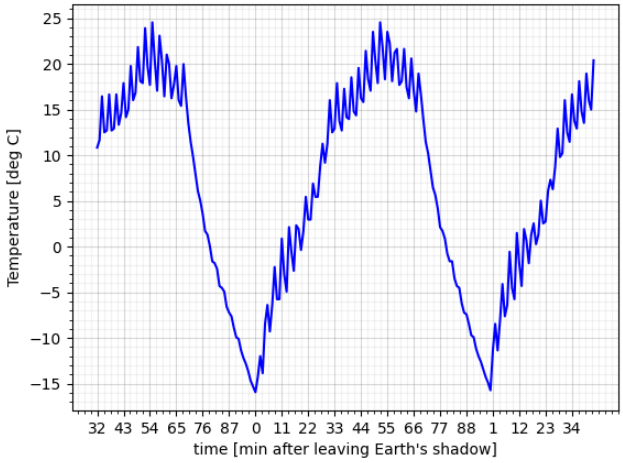


Figure 4.3: FUNcube-1 +Y panel temperature showing a ±3 °C spin-imprinted ripple on top of the 90 min orbit waveform (from [45]).

Sensor-Level Interpretation

The selected temperature channels are not equally sensitive to all thermal behaviour. External panels are most sensitive to illumination, attitude, surface properties, and environmental exposure. Internal nodes respond more slowly and are influenced by structural coupling and subsystem self-heating. Battery temperature is influenced by both the internal thermal environment and EPS-specific safety risks. This means that a similar temperature deviation may suggest different concerns depending on whether it appears on an external panel, an internal node, or the battery.

Table 4.3 summarizes the nominal temperature behaviours relevant to the AD scope, their expected telemetry effect, and their treatment as nominal context, residual texture, or out-of-scope mode dependence. The remainder of this chapter defines deviations from this nominal behaviour.

Table 4.3: Qualitative impact of environmental drivers on Delfi Twin temperature telemetry and how they should be treated by the AD.

Source	Expected telemetry effect	Scope implication
Orbit day-night cycle	Large, periodic heating and cooling over each orbit	Model as nominal baseline
Spin and nutation	Small ripple on top of the orbit waveform	Treat as nominal ripple or fine-scale variation
β-angle variation	Slow change in orbital mean and amplitude	Treat as seasonal context
Annual irradiance variation	Slow annual baseline modulation	Treat as slow nominal baseline variation
Operational mode changes	Context-dependent shifts in heating	Exclude except for `Nominal Mode`

4.5. Temperature Anomaly Vocabulary

In this thesis, a temperature anomaly is an observed deviation from the expected nominal thermal pattern. The cause may be a physical thermal fault, a telemetry-quality issue, a measurement artifact, changing operating context, or mismatch in the expected thermal model. The detector reports the direction, timing, magnitude, and evidence of the deviation; it does not establish physical root cause without additional telemetry and operational context.

To describe observable temperature patterns, the thesis adopts the general point, contextual, and collective anomaly taxonomy of Chandola et al. [10], together with the spacecraft-oriented Type I to Type V taxonomy of Cuéllar et al. [11]. The latter describes anomaly patterns in terms of telemetry shape, such as spikes, shifts, amplitude changes, timing changes, and abrupt pattern changes. The resulting vocabulary is summarized in Table 4.4 and illustrated in Figure 4.4.

Table 4.4: Temperature anomaly pattern vocabulary used in this thesis, aligned with the taxonomy of Cuéllar et al. [11]

Type / Pattern	Interpretation in temperature telemetry
I: Point spike	One or a few samples deviate sharply from neighbouring values.
II: Level shift	The waveform shifts upward or downward while preserving its general shape.
III: Waveform / amplitude	The orbit-scale swing changes, for example stronger panel heating or suppressed modulation.
IV: Frequency / phase	Peaks, troughs, or ripple features occur earlier, later, or at a different period.
V: Abrupt pattern change	Values remain within expected ranges, but the shape deviates for the current conditions.

The vocabulary is used later to describe whether a temperature deviation is point-like, sustained, amplitude-related, timing-related, or an in-limits pattern change. Figure 4.4 illustrates these shapes on a representative temperature waveform. The thesis-specific step is the mapping of the adopted taxonomy to Delfi Twin temperature telemetry, so that benchmark labels, detector outputs, and alert summaries use a consistent pattern vocabulary.

4.6. Temperature-Telemetry Pathologies and Scope Categories

Not all anomalous telemetry reflects a physical thermal fault. Spacecraft telemetry can also be affected by sensing errors, packet handling, calibration changes, communications losses, timestamping issues, preprocessing effects, and operational context. Section 2.5 reviewed these generic telemetry pathologies. This section maps them onto the temperature-telemetry scope used for Delfi Twin by separating non-physical telemetry behaviour into two categories: benign telemetry nuisances and telemetry-quality anomalies.

Benign telemetry nuisances are minor data effects that should not normally generate operator-facing events. Examples include small quantization effects, low-amplitude ordinary noise, isolated point spikes, and short missing-data intervals. These effects may perturb the evidence stream, but they should generally be tolerated or suppressed by later decision logic.

Telemetry-quality anomalies are abnormal behaviours of the measurement or processing chain that may warrant operator awareness because they compromise interpretation of the temperature stream. Examples include pinned values, saturation-like behaviour, and abrupt calibration steps. These events do not necessarily indicate a physical thermal fault, but they can still be operationally relevant because they reduce confidence in the affected telemetry channel.

The telemetry pathologies considered in this scope are summarized in Table 4.5. Illustrative examples are shown in Figure 4.5.

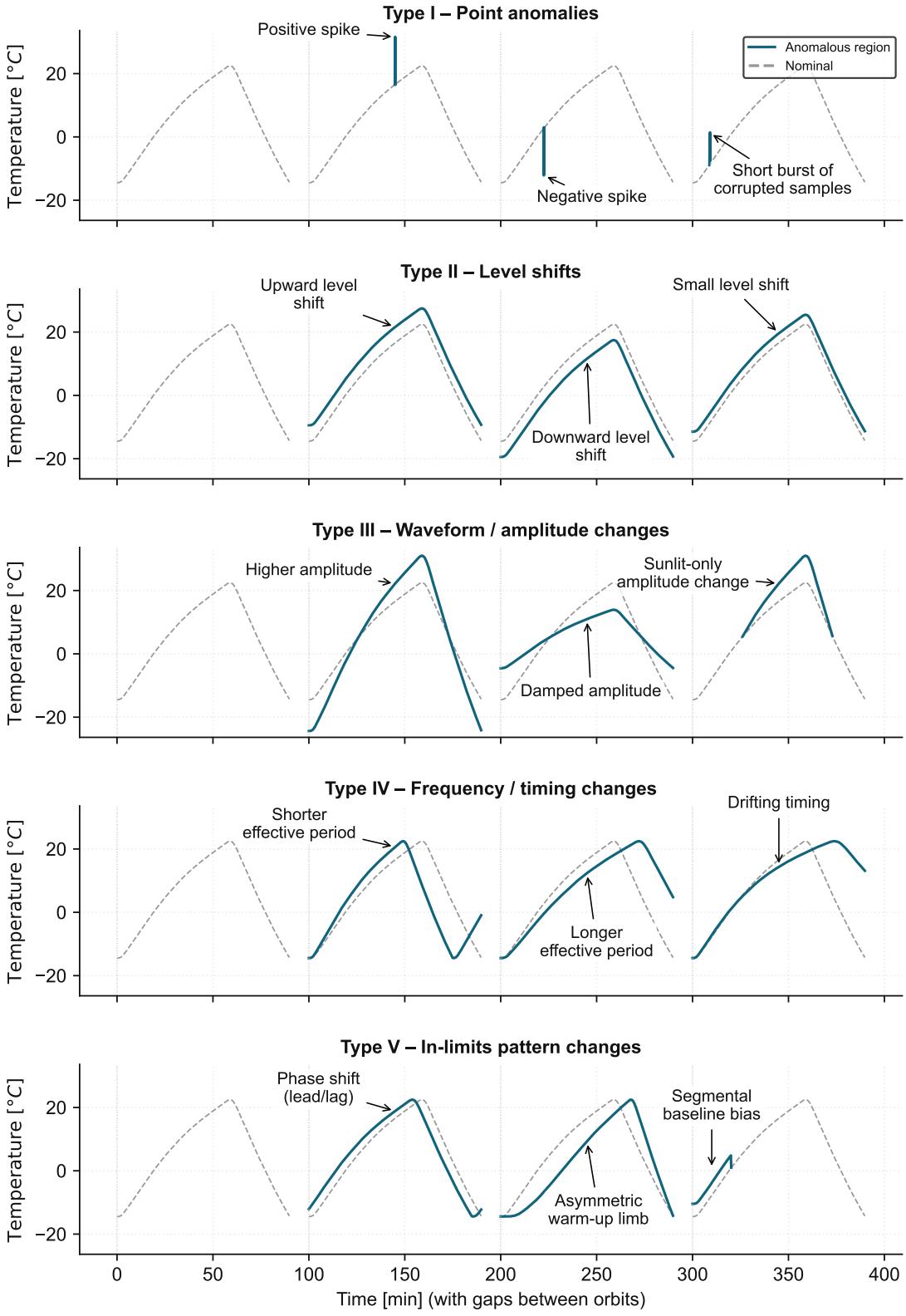


Figure 4.4: Illustrative examples of temperature anomaly pattern types used in this thesis. The dashed grey curve shows the nominal orbit-scale MCU temperature waveform (adapted from [5]), while the solid blue segments highlight anomalous data for the five anomaly types defined in [11]. The horizontal axis represents successive orbits schematically (not true time) and is intended only to illustrate the qualitative shape of each anomaly.

Table 4.5: Telemetry pathologies considered for Delfi Twin temperature AD, summarizing their operational interpretation, how they appear in telemetry, and how they can confound AD.

Subtype	Operational interpretation	Telemetry signature	Relevance for AD
		Benign nuisances	
Small quantization	Finite sensor or telemetry resolution	Small step-like changes	Can inflate derivatives and create apparent micro-jumps
Isolated spikes	Glitches or transient outliers	One-sample or few-sample excursions	Can mimic Type I anomalies, producing false events
Ordinary noise	Measurement noise	Low-amplitude jitter	Can create transient decision threshold crossings
Missing data	Downlink blackout or data dropout	Data gaps or irregular timestamps	Can break windowed scoring or prolong stale anomaly evidence
		Telemetry-quality anomalies	
Pinned values	Temporary sensor / readout freeze.	Unnaturally constant segment.	Can mimic a stable but false thermal regime
Abrupt calibration step	Sudden calibration change or conversion offset.	Sudden offset while local shape is preserved.	Can mimic a physical level shift

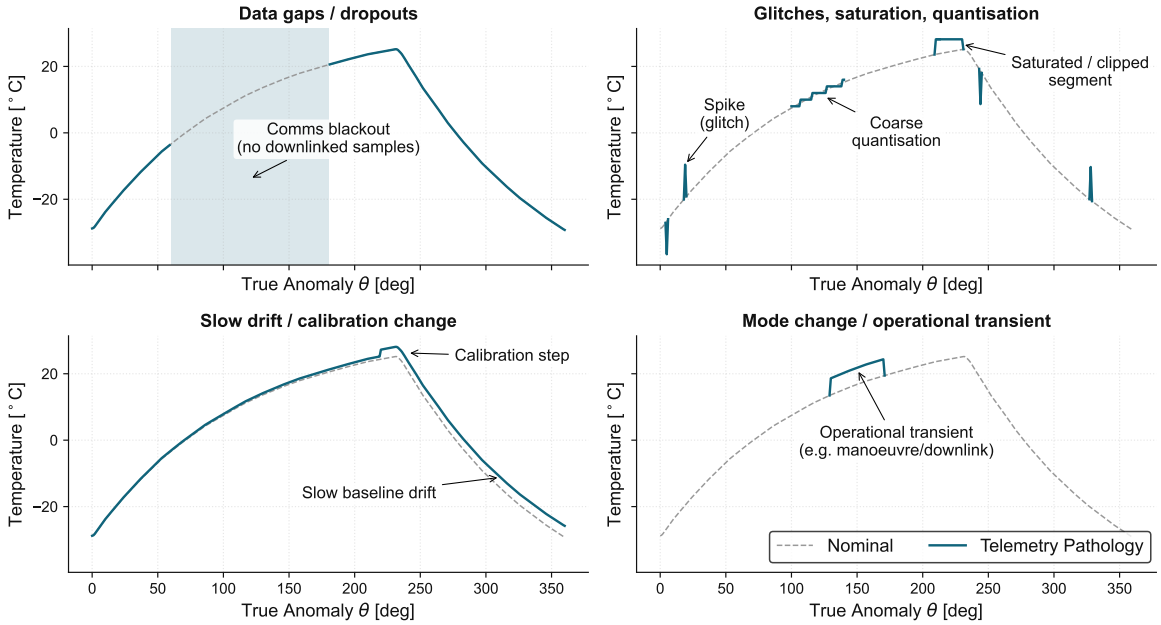


Figure 4.5: Illustrative temperature anomaly patterns used in this thesis, shown on a representative Delfi-PQ +Y panel temperature orbit waveform.

The separation between benign nuisances and telemetry-quality anomalies determines how later decision logic should respond. Benign nuisances primarily test robustness and suppression, whereas telemetry-quality anomalies may warrant alerting because they can make the affected measurement stream unreliable.

4.7. Literature-Derived Thermal Fault Families

Real labelled spacecraft datasets rarely identify the physical fault underlying a temperature anomaly. This creates a practical problem for event-level thermal alerting: the behaviours of greatest operational interest are rarely available as labelled examples for detector validation. The thesis therefore defines a literature-derived set of observable thermal fault families that can be represented in the synthetic benchmark and used to test whether the detector produces timely, evidence-based alerts for mission-relevant abnormal behaviour.

These families are mission-scoped and physically motivated, but they are not a complete Failure Modes, Effects, and Criticality Analysis (FMECA). Their purpose is to define plausible thermal behaviours observable from the selected temperature channels. A full flight FMECA would require systematic identification of failure modes, effects, and criticality across the relevant spacecraft elements, using detailed design information, mission requirements, dependability and safety objectives, environmental assumptions, and mitigation planning [19].

The retained fault families satisfy three criteria: they are plausible for a Delfi Twin-class picosatellite, at least partially observable in the selected temperature channels, and useful for evaluating event-level temperature AD across distinct thermal deviation patterns.

Delfi Twin has largely passive thermal control, with limited heating available for the battery and selected electronic components [54]. Its temperature behaviour is therefore mainly shaped by internal subsystem dissipation, external orbital heating and cooling, attitude and spin-dependent illumination, thermal coupling through interfaces and surfaces, and battery thermal limits [25, 38, 33]. Based on these thermal drivers, this thesis defines six thermal fault families for evaluating the AD pipeline. Table 4.6 introduces these families and summarizes their expected thermal effect and operational relevance.

Table 4.6: Thermal fault families considered in this work.

Fault family	Main affected nodes	Expected temperature signature	Anomaly types	Ref.
		Internal dissipation of subsystems		
Heater control failure	Battery and nearby internal nodes	Heater stuck ON or OFF produces sustained hotter or colder battery behaviour, often with shifted orbit mean and modified eclipse recovery.	Type II, III	[72]
Self-heating anomaly	OBC, EPS, COMMS, payload, battery	Unexpected subsystem duty cycle produces a localized increase or decrease in mean temperature during activity windows.	Type II, III, V	[38]
		External orbital environment		
Attitude or illumination anomaly	External panels first; internal nodes later	Off-nominal attitude, spin, or precession changes panel amplitudes, opposing-face gradients, and ripple amplitude or period.	Type III, IV, V	[47]
		Thermal impedance and surface properties		
Surface / coating degradation	External panels, coupled internal nodes	Thermo-optical ageing of coatings, panels, or insulation causes slow changes in orbit mean or thermal envelope.	Type II, III	[30]
Interface degradation	Coupled internal and external nodes	Increasing thermal resistance changes thermal gradients, time constants, and amplitude coupling between nodes.	Type III, V	[24]
		Battery safety limits		
Battery over-temperature	Battery primary; nearby nodes secondary	Incipient runaway produces a rapid nonlinear battery temperature rise, potentially followed by saturation or telemetry loss.	Type II, III, V	[33]

These families are retained because they are physically plausible for a Delfi Twin-class spacecraft, observable to different degrees in the selected temperature channels, and operationally relevant. HEAT, MODE, and RUN affect thermal safety or subsystem availability directly. SPIN captures attitude-related thermal consequences. DEG and COND represent slower degradation processes that can shift the nominal thermal baseline and reduce confidence in a fixed thermal model over mission life.

Table 4.6 defines the thermal fault scope used to guide benchmark design and detector evaluation. Later quantitative evaluation focuses on behaviours that can be represented as bounded labelled events with clear onset, offset, affected channel, and alert-worthiness. Fault families that are less suitable for bounded event injection still inform robustness testing, predictor-validity analysis, and future operational extensions.

HEAT: Heater Control Failure

Heaters are a standard element in spacecraft thermal control, used primarily on the battery, but can also be present on select components to keep electronics within their operational limits [72]. A heater stuck `ON` can overheat the battery and nearby electronics, reducing the component lifetime and risking damage. A heater stuck `OFF` can leave the battery too cold, leading to poor charging, reduced usable capacity, and possible mission loss in extreme cases. While Delfi-PQ was flown without a battery heater due to power constraints, Delfi Twin will include a dedicated heater [54].

For Delfi Twin-class temperature telemetry, the battery is the most relevant node for heater-related behaviour, with possible weaker coupling to nearby internal components. In the anomaly vocabulary, a heater-control failure is expected to appear primarily as a Type II level shift or a sustained interval-type deviation. Depending on whether the fault corresponds to excess or missing heating, the temperature deviation may be positive or negative. HEAT is therefore retained as a physically meaningful and observable thermal fault family within the scope of this thesis.

MODE: Under/Excess Subsystem Self-Heating

Subsystems such as COMMS, EPS, OBC, or payload hardware can dissipate more or less power than expected because of mode errors, duty-cycle anomalies, failed activation, or unintended activity. This fault family is analogous to heater-control failure, except that the heat source is the affected subsystem rather than a dedicated heater. For example, MinXSS-2 [44] observed a temperature increase of more than 30 °C in the transceiver and neighbouring boards during a 10 min transmission window. Reduced or missing self-heating can also be operationally relevant. On Delfi-PQ, the battery operated at a lower temperature than expected, aborting transmission attempts until the system warmed up passively [70]. Subsystem self-heating faults therefore appear as activity-related temperature mismatches: excess dissipation produces a local hotter-than-expected shift, while missing expected dissipation produces a colder-than-expected response during the relevant activity period.

This family is contextual because self-heating depends on operational state. The same temperature rise may be nominal during a high-power activity and abnormal during a quiet mode. For this reason, mode-aware detection is an important future extension. In the present thesis, MODE is evaluated only within the controlled `Nominal Mode` scope.

For the selected temperature channels, self-heating anomalies are most directly relevant to internal nodes, such as the MCU/OBC, with possible coupling to the battery or nearby structures. In waveform terms, MODE can appear as a sustained Type II level shift, a bounded excursion, or a contextual pattern change. It is retained in the scope because it represents a common way in which subsystem activity can become visible in temperature telemetry.

SPIN: Spin, Attitude, or Illumination Anomaly

The thermal state of the spacecraft also depends on how long each face sees Sun versus shadow. Therefore, changes in the attitude and spin rate will modify which spacecraft surfaces are illuminated and for how long. Low-spin or poor attitude pointing can produce large hot-cold gradients and orbit-scale swings on external faces, whereas high-speed spin reduces those swings and better equalizes the temperature gradients between opposite faces [47]. These effects are expected to appear first in external panel temperatures, especially through changes in orbit-scale amplitude, opposing-panel gradients, or spin-ripple behaviour.

For Delfi Twin, large attitude or spin anomalies are treated as rare ADCS-related faults with thermal consequences, rather than primary thermal failures. A realistic SPIN event would require modelling how the attitude change alters illumination and thermal response. Without this coupling, a temperature-only representation would test the detector's response to a thermal signature, but would not validate detection of the underlying attitude-related fault.

SPIN is therefore retained as a relevant attitude-related mechanism with thermal consequences, but it is not treated as a primary bounded thermal-fault event class in the Part I scope. Spin and nutation are treated primarily as nominal thermal variation unless additional attitude-aware evidence is available.

DEG: Surface or Coating Degradation

Coatings, panels, and insulation have thermo-optical properties that determine how much heat is absorbed from external sources and emitted to space. These properties can change over mission life due to ultraviolet exposure, atomic oxygen, contamination, or outgassing [30]. Such changes can alter absorbed solar energy and emitted thermal radiation, producing gradual shifts in orbit-mean temperature, orbit-scale amplitude, or the relationship between external and internal temperature nodes.

DEG is included because it represents a physically plausible source of long-term thermal change. For a temperature-based detector, this matters because surface degradation can shift the nominal thermal baseline and reduce confidence in a predictor trained on earlier mission behaviour. It therefore provides a useful fault family for reasoning about predictor validity, long-term drift, and the distinction between abrupt alert-worthy events and slowly evolving thermal changes.

COND: Thermal Interface or Conductivity Degradation

Thermal interface materials (Thermal Interface Material (TIM)) are used at contact points between components and heat-rejection paths to reduce contact resistance and improve heat transfer through the spacecraft structure [59]. Like thermo-optical coatings, these materials may be affected by the space environment, including ultraviolet exposure, atomic oxygen interactions, contamination, and outgassing. Changes in thermal conductivity or contact resistance can alter how efficiently heat flows between a source and a sink.

COND is included because it represents a physically plausible change in thermal coupling rather than only a change in heat generation. Observable effects may include altered thermal gradients, changed lag between external and internal nodes, weaker or stronger coupling to orbital heating, or local overheating when heat is no longer conducted away efficiently. For a temperature-based detector, this fault family is relevant because it can change multichannel thermal relationships and reduce confidence in a fixed nominal thermal model over mission life.

RUN: Battery Over-Temperature or Incipient Thermal Runaway

Lithium-ion (Li-ion) batteries are widely used in spacecraft because they provide high energy density at low mass, but their performance, lifetime, and safety are temperature-sensitive. One high-consequence battery failure mode is thermal runaway, in which elevated temperature can initiate heat-generating reactions that further increase cell temperature [33]. Battery protection must therefore remain a hardware and system-level safety function, especially because external temperature sensing may lag internal cell behaviour.

RUN is retained as a severe thermal-fault family because it represents a rapid, high-magnitude positive temperature deviation with high alert urgency. A telemetry-based temperature AD function cannot replace battery protection circuitry or guarantee intervention before a critical failure. Its role is narrower: to report timely, interpretable thermal evidence from the available battery temperature sensor when the measured response departs strongly from the expected nominal pattern.

4.8. Scope Boundaries

The anomaly scope is deliberately bounded. The thesis evaluates whether temperature telemetry can support explainable on-board anomaly awareness under MCU-class constraints. It does not attempt full autonomous fault diagnosis, full Fault Management (FM) integration, all-mode thermal validation, or exhaustive spacecraft FMECA. It also does not assume that temperature telemetry alone can uniquely identify root cause.

The main scope boundaries are summarized in Table 4.7.

Table 4.7: Scope boundaries for the on-board temperature AD prototype.

Design aspect	In scope	Out of scope
Telemetry source	Temperature housekeeping channels	Full voltage-current-temperature AD
Operating mode	`Nominal Mode`	Full mode-aware flight operation
Alert output	Event alerts and detector-faithful evidence	Autonomous root-cause diagnosis
Fault classes	Selected thermal faults and telemetry-quality anomalies	Exhaustive spacecraft FMECA
Evaluation basis	Synthetic labelled events anchored to mission-relevant behaviour	Flight-certified validation

The key boundaries are temperature-only input, `Nominal Mode` evaluation, detector-faithful evidence rather than root-cause diagnosis, and synthetic labelled evaluation rather than flight-certified validation. Voltage, current, mode flags, and other status telemetry remain important for diagnosis and future context-aware detection, but they are not primary detector inputs here.

These boundaries define how later results should be interpreted. They also explain why the thesis separates synthetic benchmark evaluation, embedded verification, FUNcube real-telemetry stress testing, and deployment-readiness interpretation rather than treating any single evidence source as sufficient for flight validation.

4.9. Chapter Summary

This chapter completed the telemetry and anomaly-scope foundation for the thesis. It identified EPS temperature telemetry as the first detector target because it is physically meaningful, available in housekeeping data, compatible with lightweight expected-behaviour modelling, and suitable for evidence-based alerting. It then defined the nominal thermal behaviours that should be modelled or tolerated: orbit-scale sunlight and eclipse cycles, spin and nutation texture, beta-angle variation, annual irradiance modulation, and operational mode changes.

The chapter adopted a Type I to Type V anomaly-pattern vocabulary for temperature telemetry, separated benign telemetry nuisances from telemetry-quality anomalies, and defined six literature-derived thermal fault families: HEAT, MODE, SPIN, DEG, COND, and RUN. These families define the thermal behaviours considered by the thesis, but they do not constitute a full FMECA. Later benchmark construction focuses on behaviours that can be represented as reproducible labelled events, while the remaining families inform robustness testing, predictor-validity analysis, and future extensions.

Together with the operational and embedded requirements derived in Chapter 3, this telemetry and anomaly scope completes the Part I foundation. Chapter 5 synthesizes Chapters 2 to 4 into the formal research questions, thesis contributions, and research plan.

5

Research Questions and Thesis Plan

Chapters 2 to 4 established the literature and scope foundation for this thesis. Chapter 2 reviewed spacecraft Anomaly Detection (AD), telemetry, benchmarks, embedded deployment, score-to-event methods, and explainable alerting. Chapter 3 translated Fault Management (FM) and embedded-deployment literature into specific operational, timing, and resource requirements for the Delfi Twin case study. Chapter 4 defined the temperature-telemetry scope, nominal thermal behaviour, anomaly semantics, telemetry pathology, and thermal fault families considered in the thesis.

Together, these chapters show that on-board thermal AD cannot be reduced to selecting an anomaly-scoring algorithm. The deployment problem also requires an operational role, observable telemetry, nominal-behaviour assumptions, a defensible ground-truth strategy, score-to-event logic, compact alert content, and embedded feasibility. The thesis therefore treats AD as a complete residual-to-alert workflow rather than as a standalone scoring task.

This chapter closes Part I by translating that foundation into the formal research objective, research questions, thesis contributions, and execution plan. The detailed design, implementation, evaluation, and operational interpretation are presented in Parts II to IV.

5.1. Part I Synthesis and Research Gap

The literature reviewed in Chapter 2 identifies a deployment gap in spacecraft telemetry AD. Much of the literature emphasizes anomaly scoring, classification, or offline benchmark performance. For on-board use, however, the score is only one part of the operational product. The detector must also decide whether the evidence forms a reportable event, summarize that event compactly, and preserve the evidence needed for ground review.

Chapter 3 instantiated this problem for the Delfi Twin case study. The on-board function is scoped as advisory early awareness: it should detect suspected temperature events, characterize the evidence, and package compact alerts. It should not perform autonomous diagnosis, select the operational response, or manage model updates without ground involvement. The same chapter derived the timing and embedded constraints used later in the thesis: 15 s telemetry ingestion, a 60 s alert-frame target, compact alert packets, causal operation, and Microcontroller Unit (MCU)-class resource budgets.

Chapter 4 completed the scope foundation by defining what the detector may observe and what it should not overclaim. Temperature telemetry was selected because it is physically meaningful, widely available as housekeeping data, slowly varying, and suitable for lightweight expected-behaviour modelling. The same chapter separated nominal thermal structure from benign nuisance behaviour, telemetry-quality anomalies, and physical thermal-fault families.

The resulting research gap is an integration gap. Existing work provides anomaly-scoring methods, telemetry datasets, embedded constraints, event logic, and explainability concepts, but these elements are rarely connected into a complete pathway for real-time, explainable, on-board thermal event alerting in downlink-constrained missions. Delfi Twin provides the MCU-class case study used to make that pathway concrete. Table 5.1 summarizes how the Part I findings motivate the thesis plan.

Table 5.1: Part I synthesis: literature-derived findings, remaining gaps, and implications for the thesis plan.

Part I finding	Remaining gap	Implication for the thesis plan
Spacecraft AD literature often emphasizes anomaly scoring or classification	Point-wise scores alone do not define operational alert behaviour	The thesis must include score-to-event logic and compact alert content, not only anomaly scoring.
Downlink-constrained missions require selective telemetry reporting	Continuous full-resolution telemetry may not be available to operators in time	The thesis must evaluate whether temperature deviations can be converted into compact, event-level alerts suitable for prioritized downlink.
Resource-constrained small spacecraft impose compute, memory, power, bus, and packet-size limits	Offline detection performance does not establish embedded feasibility	The thesis must verify timing, memory use, packet size, and resource feasibility on MCU-class hardware.
Real spacecraft telemetry contains gaps, artifacts, drift, and mode-dependent behaviour	Detector outputs may be misleading without observability assessment and event-lifecycle logic	The thesis must define telemetry-pathology categories, nominal-behaviour assumptions, and lifecycle handling for bounded events.
Public spacecraft datasets rarely provide fault-level and alert-worthiness labels	Event-level thermal alerting cannot be fully validated from existing labelled datasets alone	The thesis requires a controlled labelled benchmark, kept distinct from real-telemetry stress testing.
Temperature telemetry is physically meaningful but context-dependent	Nominal thermal variation can resemble anomaly evidence	The thesis must define expected nominal thermal behaviour, construct an expected-temperature baseline, and evaluate residual evidence.
Expected-temperature predictors can become stale or mismatched	Residual evidence may reflect predictor mismatch rather than abnormal spacecraft behaviour	The thesis must evaluate predictor-mismatch sensitivity and the conditions under which adaptive correction is useful or unsafe.
Explainable alerts are needed for operator assessment	A compact on-board alert cannot provide definitive root-cause diagnosis	The thesis must report detector-faithful evidence while leaving diagnosis, response selection, and model updates to ground operators.

These findings motivate the research objective and research questions below.

5.2. Research Objective

Based on the Part I synthesis, the objective of this thesis is:

Research Objective

To design, implement, and evaluate a deployment-oriented, explainable, on-board temperature AD pipeline for Delfi Twin as an MCU-class case study, while developing a reusable workflow and evaluation protocol for assessing event-level thermal alerting in downlink-constrained space missions.

The objective has two parts. The first is technical: to construct the data, predictor, decision logic, alert concept, and embedded implementation needed for an on-board temperature AD prototype. The second is methodological: to define an evaluation workflow that separates controlled synthetic event-level validation, embedded feasibility testing, predictor-mismatch analysis, and qualitative real-telemetry stress testing. This separation is required because each evidence source supports a different type of claim.

The thesis therefore does not aim to produce a flight-qualified AD system. It aims to determine whether the residual-to-alert concept is technically plausible, operationally meaningful, and resource-feasible on representative MCU-class hardware, and to identify the additional validation required before flight deployment or autonomous beacon-style use.

5.3. Research Questions

The research questions follow the deployment pathway established in Part I. They progress from operational formulation and telemetry scope, through benchmark design and pipeline selection, to embedded implementation and operational readiness.

Research Questions

RQ-1: Operational formulation and requirements
How should on-board thermal AD be operationally formulated for a downlink-constrained mission, and what real-time, alerting, responsibility, and embedded constraints follow when this role is instantiated for Delfi Twin?

RQ-2: Telemetry suitability and anomaly semantics
Which temperature telemetry channels and anomaly classes are suitable for a first deployment-oriented on-board AD prototype, and how should nominal thermal behaviour, telemetry pathologies, and physical thermal faults be distinguished?

RQ-3: Benchmark and ground-truth protocol
How can a controlled and reproducible benchmark be constructed to evaluate event-level thermal AD when real spacecraft telemetry lacks reliable fault-level and alert-worthiness labels?

RQ-4: Pipeline selection, configuration, and event formation
Which end-to-end AD pipeline is most appropriate for the selected telemetry, anomaly scope, and deployment constraints, and how should residual evidence be converted into bounded, explainable alerts?

RQ-5: Embedded implementation and performance
Can the selected pipeline be implemented causally on STM32L4-class hardware while satisfying the timing, memory, packet-size, and resource constraints derived for the case-study mission?

RQ-6: Operational value and deployment readiness
What operational value does the prototype provide for downlink-constrained thermal anomaly awareness, what are its main limitations, and what further validation is required before flight deployment or autonomous beacon-style use?

5.4. Thesis Contributions

This thesis makes three main contributions. The contributions lie in selecting, combining, implementing, evaluating, and interpreting established primitives as a coherent on-board thermal alerting pathway. Individual elements such as residual scoring, threshold evidence, cumulative evidence, persistence, hysteresis, detector-faithful explanation, and embedded timing measurement are established concepts. The contribution is their integration into a mission-aligned workflow for event-level thermal alerting in a downlink-constrained setting, using Delfi Twin as the MCU-class case study.

Main Thesis Contributions

C-1: Deployment-oriented workflow for event-level thermal alerting
The thesis formulates on-board temperature AD as an advisory early-awareness function for downlink-constrained missions, instantiated through the Delfi Twin case study. It defines the FM responsibility split, real-time and embedded constraints, telemetry scope, anomaly semantics, synthetic evaluation protocol, and interpretation limits required for deployment-oriented assessment.

C-2: Explainable residual-to-event detector and compact alert packets
The thesis develops a lightweight residual-based thermal anomaly alerting pipeline. Expected-temperature prediction, normalized residuals, instantaneous and cumulative evidence, persistence, hysteresis, quiet reset, transient suppression, and gap termination are combined to form bounded detector events. These events are encoded into compact, detector-faithful alert packets that report deviation magnitude, thermal direction, duration, timing, and contributing evidence without claiming on-board root-cause diagnosis.

C-3: Embedded verification, robustness, and deployment-readiness assessment
The thesis implements the pipeline on STM32L4-class hardware and verifies timing, memory use, packet size, and resource feasibility. It evaluates baseline and ablation cases, tests robustness under predictor mismatch, applies the system to FUNcube-1 telemetry as qualitative real-telemetry stress cases, and identifies the limitations that must be addressed before flight deployment or autonomous beacon-style use.

5.4.1. Boundary Between Literature Primitives and Thesis Contribution

Table 5.2 clarifies the boundary between prior primitives and the thesis contribution. The thesis does not claim novelty in the individual decision primitives or AD mechanisms. Its contribution is the case-study-specific integration, implementation, evaluation, and operational synthesis of these primitives into an embedded residual-to-alert workflow.

Table 5.2: Known literature primitives, their integration in this thesis, and the contribution enabled.

Known primitive from literature	Integration in this thesis	Contribution enabled
Low-power CubeSat thermal monitoring precursor	Maununen [45] motivates on-board thermal anomaly monitoring on Delfi-PQ-class hardware.	Residual-to-event alerting workflow with gap semantics, alert packets, predictor-validity analysis, and STM32L4 replay.
FM responsibility split	The on-board function is limited to advisory detection, event formation, and compact alert packaging; diagnosis, response selection, and adaptation remain ground responsibilities.	Operationally bounded deployment workflow.
Telemetry observability and anomaly semantics	Temperature telemetry is separated into nominal behaviour, benign nuisances, telemetry-quality anomalies, and physical thermal faults.	Mission-scoped anomaly vocabulary for event-level thermal alerting.
Ground-truth limitations in spacecraft telemetry	A labelled synthetic benchmark supports controlled event-level metrics, while FUNcube-1 supports qualitative real-telemetry stress testing.	Evaluation protocol with explicit interpretation limits.
Generic labelled synthetic telemetry primitives	Synthetic anomaly primitives from Schefels et al. [60] inform the injected event design.	Mission-specific thermal benchmark with event-level labels and claim boundaries.
Expected-temperature prediction and residual scoring	A lightweight predictor produces signed and normalized residual evidence for each monitored thermal node.	Measurement-based thermal anomaly evidence.

Continued on next page

Table 5.2: Known literature primitives, their integration in this thesis, and the contribution enabled. Continued.

Known primitive from literature	Integration in this thesis	Contribution enabled
Threshold evidence, cumulative evidence, and lifecycle logic	Threshold evidence, cumulative evidence, persistence, hysteresis, transient suppression, quiet reset, and gap termination are combined into a causal residual-to-event lifecycle.	Bounded anomaly events rather than point-wise scores.
Detector-faithful explainability	Alert packets report quantities used by the detector, including affected node, event timing, direction, peak evidence, reason flags, and clearance reason.	Compact explainable alerts without on-board root-cause diagnosis.
Embedded resource budgeting and MCU feasibility testing	The selected pipeline is implemented and replayed on STM32L4-class hardware with measured timing, memory use, packet size, and resource behaviour.	Prototype evidence for MCU-class feasibility.
Robustness and deployment-readiness assessment	Predictor mismatch, adaptive correction, and real-telemetry stress cases are used to identify prototype validity and remaining deployment limitations.	Deployment-readiness assessment for continued development.

5.5. Deployment-Oriented Workflow

The research questions are addressed through the workflow shown in Figure 5.1. The workflow is staged deliberately. The mission role and constraints define what the on-board function is allowed to do. The telemetry and anomaly-scope analysis defines what can be observed, what should be treated as nominal, and which temperature deviations are meaningful to evaluate. These foundations support the benchmark and ground-truth protocol, which then enable pipeline selection, embedded implementation, evaluation, and deployment-readiness assessment.

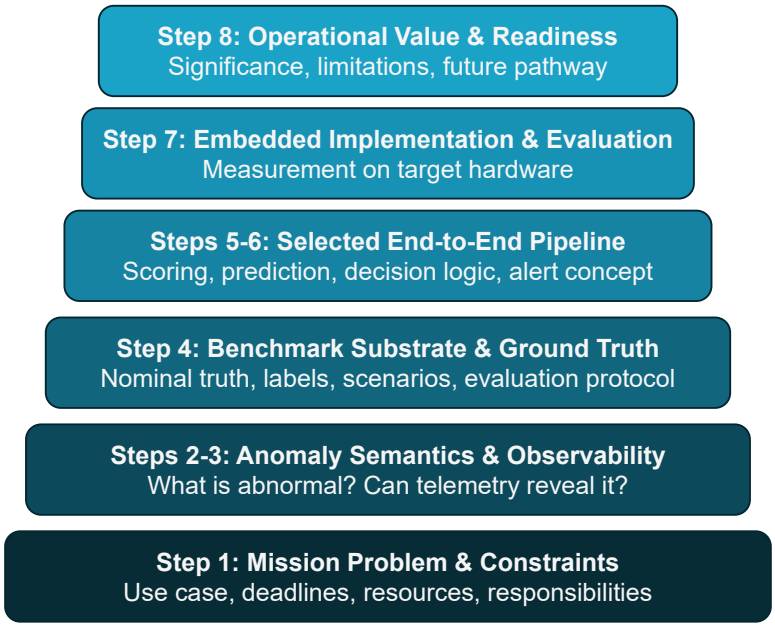


Figure 5.1: Stacked representation of the deployment-oriented workflow, emphasizing that embedded implementation and verification depend on the foundations established in the lower workflow layers.

The workflow keeps the thesis focused on event-level thermal alerting. Later chapters do not evaluate an anomaly score in isolation. They evaluate the path from temperature telemetry, expected-temperature prediction, and residual evidence to compact, explainable, resource-feasible alerts, using Delfi Twin as the MCU-class case study.

5.6. Research Plan and Thesis Structure

Table 5.3 maps the workflow steps to research questions, thesis outputs, and chapters.

Table 5.3: Research plan mapping workflow steps to research questions, outputs, and thesis chapters.

Workflow step	RQ(s)	Main output	Chapter(s)
1. Establish mission problem and deployment constraints	RQ-1	Operational role, timing requirement, resource budgets	Chapter 3
2. Define mission-relevant anomalies and success criteria	RQ-2	Anomaly vocabulary, telemetry-pathology categories, thermal fault scope	Chapter 4
3. Establish telemetry observability and nominal behaviour	RQ-2	Temperature-channel justification, nominal thermal behaviour, observability limits	Chapter 4
4. Design dataset and ground-truth protocol	RQ-3	Clean nominal reference and labelled synthetic benchmark	Chapters 6, 7
5. Screen and down-select the end-to-end pipeline	RQ-4	Selected residual-to-event architecture	Chapter 8
6. Tune and evaluate the selected pipeline	RQ-4, RQ-6	Predictor, decision configuration, evaluation results, robustness analysis	Chapters 9, 10, 12
7. Implement and verify under embedded constraints	RQ-5	Embedded firmware, timing, memory, packet, and replay verification	Chapters 11, 12
8. Assess operational significance and future pathway	RQ-6	FUNcube stress tests, operational value, limitations, go/no-go discussion	Chapters 13, 14, 15

The remainder of the thesis executes this plan. Part II develops the benchmark and detector design: Chapter 6 constructs a clean nominal thermal reference, Chapter 7 builds the labelled synthetic benchmark, Chapter 8 selects the pipeline architecture, Chapter 9 defines and verifies the expected-temperature predictor, and Chapter 10 defines the residual decision and explainable alert layer. Part III implements and evaluates the pipeline: Chapter 11 presents the embedded implementation, Chapter 12 reports results, and Chapter 13 applies the detector to FUNcube telemetry as a qualitative stress test. Part IV interprets operational value, limitations, deployment readiness, and final conclusions.

Part II:

Data and Method Design for the End-to-End Pipeline

Building on the problem framing and scope established in Part I, this part develops the benchmark and residual-based detection framework used in the thesis. It constructs the nominal and labelled datasets, selects the prediction and scoring approach, and defines the residual-to-alert pipeline that is later implemented and evaluated.

6

Nominal Thermal Truth Generation

Part I established the nominal thermal behaviours that must be treated as expected spacecraft behaviour rather than as faults: orbit-scale heating and cooling, seasonal illumination variation, annual irradiance modulation, spin-imprinted thermal ripple, and low-level measurement noise. This chapter implements those literature-derived drivers as a compact clean reference dataset for controlled anomaly injection. The resulting dataset is used as the nominal substrate for the synthetic benchmark in Chapter 7.

The purpose of this chapter is to construct the clean nominal reference needed for later controlled evaluation. It does not propose a flight-certified thermal simulator, nor does it claim to reproduce the full thermal dynamics of Delfi Twin. Instead, it builds a repeatable, physically plausible, multichannel nominal temperature reference with a known time base, known drivers, and no injected anomalies. This allows later chapters to inject controlled telemetry pathologies and thermal faults with explicit onset, offset, affected channel, anomaly class, and alert-worthiness labels.

The output is a five-year clean multichannel thermal truth dataset for the battery, four solar panels, and Microcontroller Unit (MCU) temperature nodes. This dataset is passed forward to Chapter 7, where labelled telemetry pathologies and thermal fault signatures are injected.

Deployment-Oriented Workflow Steps

3. Establish telemetry observability and nominal behaviour
Justify that the selected telemetry channels can plausibly reveal the target anomalies, and characterize the telemetry issues that must be tolerated (e.g. gaps, resets, quantization, drift)

4. Design the dataset and ground-truth protocol
Specify the data features, event label semantics (onset/offset), scenario suite, and train/validation/test splits that ensure result significance and reflect operational conditions.

6.1. Purpose and Scope of the Nominal Baseline

The clean nominal baseline has three roles in the thesis. First, it provides a fault-free reference into which controlled deviations can be injected. Second, because the baseline is fixed and reproducible, it allows detector configurations to be developed, tuned, and compared on the same nominal thermal conditions. Third, it defines the expected-temperature structure used later to evaluate residual-based anomaly scoring.

This baseline is required because real spacecraft telemetry alone cannot provide the controlled ground truth needed for this thesis. Downlinked telemetry may contain gaps, artifacts, unlabelled anomalies, and incomplete operational context. Even when real telemetry is available, it rarely provides event-level labels that specify the physical class, affected channel, onset, offset, and expected operational interpretation of a thermal anomaly. Section 2.6 therefore justified synthetic injection as the quantitative evaluation strategy, while Chapter 4 identified the nominal thermal behaviours that the detector should model or tolerate.

The resulting clean truth preserves the main nominal structures needed for residual-based detector testing: orbit-scale temperature variation, slow environmental modulation, spin-imprinted ripple, and low-level measurement noise. It should therefore be interpreted as a controlled reference for anomaly injection and detector evaluation, not as a flight-certified thermal simulator, a substitute for thermal-vacuum validation, or a calibrated flight predictor.

6.2. Nominal Thermal Truth Generation Workflow

The generation workflow is shown in Figure 6.1. It begins with flight telemetry from the selected temperature nodes and first applies basic pre-processing to obtain cleaned, time-ordered temperature series. These data are then reduced to phase-binned orbit medians, which capture representative orbit-scale heating and cooling behaviour while suppressing short-term noise and irregular sampling effects. Compact templates are fitted to these phase-binned medians to provide a low-complexity description of the nominal orbit-scale waveform.

The fitted templates are then augmented with nominal environmental effects, including slow seasonal eclipse-geometry variation, annual irradiance modulation, spin-imprinted ripple, and low-level measurement noise. The final output is a clean multichannel truth dataset at the telemetry cadence used throughout the thesis. This dataset contains no injected anomalies and is passed forward to Chapter 7 as the substrate for labelled observation-layer and thermal-fault injections.

Figure 6.1: Nominal thermal truth generation workflow used to produce the clean multichannel reference for later anomaly injection.

6.3. Channels, Time Base, and Synthetic Orbit Assumptions

The clean reference is generated for the temperature nodes selected in Section 4.2: the battery, solar panels, and the MCU. These nodes were chosen to span thermally meaningful internal and external locations while remaining relevant to both physical thermal faults and telemetry-quality anomalies.

The dataset is generated at the native housekeeping cadence used throughout the detector design. A fixed sample period is used so that persistence, decision latency, Time to Awareness (TTA), and embedded replay timing can be defined consistently in later chapters. The resulting time base is a controlled benchmark axis, not a reconstruction of a specific flight telemetry timeline.

The orbital assumptions are chosen to provide a controlled nominal reference, not a full reconstruction of flight dynamics. They preserve the orbit-scale and slow environmental structure needed for detector development, while avoiding claims about the exact attitude, illumination, operational mode, or thermal state of a flight spacecraft. Table 6.1 lists the generated nodes, while Table 6.2 summarizes the orbit and time-base assumptions used for nominal truth generation.

Table 6.1: Temperature nodes included in the nominal thermal truth dataset.

Node	Interpretation
Batt	Battery
Panel Yp	+Y solar panel
Panel Ym	-Y solar panel
Panel Xp	+X solar panel
Panel Xm	-X solar panel
MCU	MCU/OBC

Table 6.2: Orbit and time-base assumptions used for nominal thermal truth generation.

Parameter	Mission value	Model value used
Orbit type	Near-circular SSO	Circular SSO
Altitude h	525 km	525 km
Inclination i	97.4°	98°
Orbit period T_{orbit}	95.13 min	95 min
LTAN	10.5 h, 11.0 h and 11.5 h	10.5 h

6.4. Representative Orbit-Scale Template Construction

The dominant component of the clean nominal reference is the orbit-scale temperature waveform. Section 4.4 identified orbit-scale sunlight and eclipse cycling as nominal behaviour that should not be treated as anomalous. This section defines how that behaviour is represented in the benchmark.

For each node, cleaned flight-derived Delfi-PQ temperature samples are aligned by orbital phase and reduced to phase-binned median curves, as illustrated in Figure 6.2a. This produces a representative orbit-scale shape while reducing the influence of isolated artifacts, incomplete samples, and local disturbances. Median curves provide a robust estimate of the central orbit-scale thermal trend by reducing sensitivity to residual outliers, unequal data density across phase, seasonal vertical spread, and small ripple/noise components. The resulting phase-binned profiles are then approximated using a compact two-branch heating and cooling template.

The branch-template form is selected because the clean reference must satisfy two competing requirements. It should remain physically interpretable enough to resemble the smooth heating and cooling behaviour expected in small-satellite temperature telemetry, while staying compact enough for later residual-based prediction and embedded implementation. The template is therefore not presented as a new general spacecraft thermal model. Its role is to provide a repeatable expected-temperature waveform for benchmark construction.

A stretched-exponential form was adapted from the fitting function proposed by Bhat [5]. A standard single-timescale exponential was found to be too rigid for the multi-orbit reference because it produced transitions that were too abrupt between sunlight and eclipse, and again from eclipse back to sunlight. The stretched form provides an additional shape parameter, allowing the heating and cooling branches to represent smoother, node-dependent thermal transitions while remaining compact enough for synthetic generation and later embedded predictor development. A representative formulation is:

$$T_{\text{heat},j}(\theta) = T_0 + A_h \left(1 - e^{-r_h \theta^{p_j}}\right) \tag{6.1}$$

$$T_{\text{cool},j}(\theta) = T_{\text{sat,fall}} + \left(T_e - T_{\text{sat,fall}}\right) e^{-r_c (\theta - \theta_E)^{p_j}} \tag{6.2}$$

Here, j denotes the thermal node, θ is the orbital phase, θ_E controls the eclipse-transition location, and p_j is the stretched-exponential shape parameter. This approach has been shown to be effective for thermal modelling in the literature [55].

Where Delfi-derived phase coverage is limited, normalized FUNcube thermal curves are used as supplementary shape guidance. FUNcube is not treated as a direct thermal surrogate for Delfi Twin; it only provides additional information about plausible orbit-scale waveform behaviour in poorly constrained phase regions. The Delfi-derived phase-binned median remains the primary fitting target. The normalized Delfi and FUNcube curves for the same channel are compared in Figure 6.2b.

The final fitting objective implements this relationship by combining the Delfi-derived phase-binned target with an auxiliary shape-regularization term based on the normalized FUNcube reference curve. The fitted parameters define the node-wise orbit-scale templates used in the remainder of the clean reference generation. This regularized fitting objective is inspired by shape-regularization approaches in inverse problems [18]:

$$J = J_{\text{Delfi fit}} + \lambda_{\text{shape}} J_{\text{FUNcube shape}}. \tag{6.3}$$

Here, $J_{\text{Delfi fit}}$ is the main fitting loss against the available Delfi-derived phase-binned temperature profile. It ensures that the selected template follows the Delfi-class thermal behaviour wherever sufficient data are available. $J_{\text{FUNcube shape}}$ is an auxiliary penalty that discourages the fitted template from departing unnecessarily from the normalized FUNcube reference shape in poorly constrained phase regions. The weighting factor λ_{shape} controls the strength of this regularization: a small λ_{shape} allows the Delfi data to dominate, while a larger λ_{shape} gives more influence to the supplementary shape constraint.

The candidate branch-model variants are compared with the Delfi median curve in Figure 6.2c, and the corresponding normalized fits are shown in Figure 6.2d. The final orbit-template formulation adopted for nominal telemetry generation was the flexible two-branch stretched-exponential model, with the FUNcube-derived shape-regularization term used only where it improved stability in poorly observed phase regions. The resulting selected parameters are summarized in Table 6.3. These node-wise templates formed the orbit-scale baseline used in the later long-duration nominal synthesis.

Table 6.3: Final selected orbit-template parameters for each nominal thermal node.

Node	θ_E [°]	λ_{shape} [–]	$T_{\text{sat,rising}}$	r_{rising} [$\times 10^{-3}$]	T_0 [°C]	$dT_{0,\text{falling}}$ [°C]	p
Batt	227.16	0.15	27.76	0.171	−5.87	11.45	1.77
Panel Yp	227.16	0.35	53.10	2.575	−25.30	8.18	1.29
Panel Ym	227.16	0.35	46.69	0.919	−29.42	2.71	1.80
Panel Xp	227.16	0.35	45.28	0.959	−27.42	6.17	1.60
Panel Xm	227.16	0.35	47.65	3.708	−23.15	6.73	1.25
MCU	227.16	0.15	32.53	0.585	−14.08	6.67	1.68

The fitted templates provide the repeating orbit-scale component of the clean reference. The following section adds the slow environmental modulation, spin-imprinted ripple, and nominal noise needed to generate the long-duration multichannel thermal truth dataset.

6.5. Simplified Nominal Environmental Drivers

Section 4.4 identified several nominal drivers that can change spacecraft temperature behaviour without representing a fault. These include (i) seasonal eclipse-geometry variation, (ii) annual irradiance variation, (iii) spin-imprinted thermal texture, and (iv) low-amplitude nominal Gaussian noise. This section defines simplified mathematical drivers for including those effects in the clean benchmark.

The purpose of these drivers is not high-fidelity thermal simulation. Their purpose is to ensure that the clean reference contains the main forms of nominal variability that a residual detector must tolerate. The resulting signal therefore includes both predictable orbit-scale behaviour and smaller nominal variations that should remain below the detector's alert threshold.

6.5.1. Seasonal Eclipse-Geometry Modulation

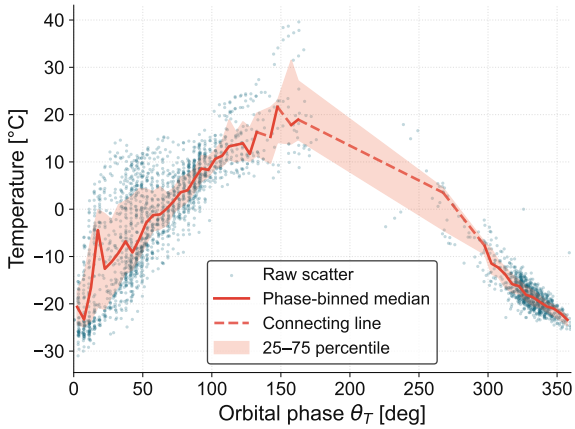
In the clean benchmark, seasonal illumination effects are represented by a compact driver that modulates the orbit-scale heating and cooling profile over time. The driver converts slowly varying illumination geometry into a time-varying eclipse-transition parameter, producing controlled changes in the duration and shape of the heating and cooling portions of the template. This exposes the detector to plausible slow nominal variation without attempting to reproduce exact orbit propagation, attitude geometry, or flight thermal dynamics.

The solar beta angle, β, is approximated as [25]:

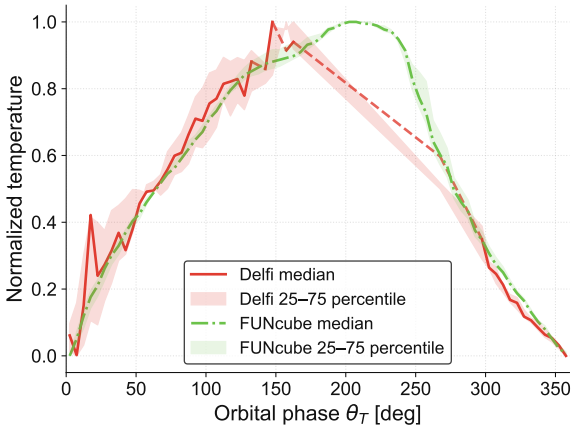
$$\beta = \sin^{-1}\left(\cos\delta_s \sin i \sin\left(\Omega - \Omega_s\right) + \sin\delta_s \cos i\right) \tag{6.4}$$

The corresponding eclipse fraction is approximated as [25]:

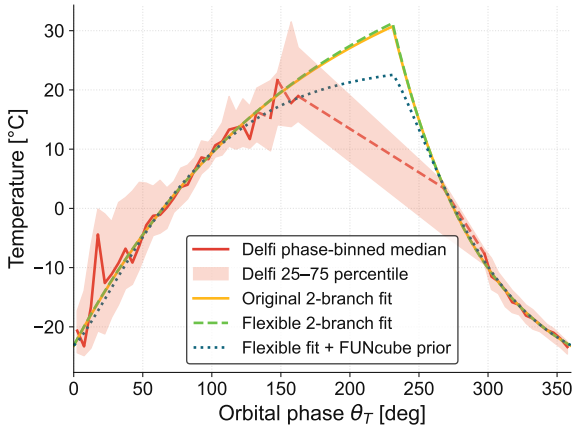
$$f_E = \frac{1}{180^\circ}\cos^{-1}\left[\frac{(h^2 + 2R_E h)^{1/2}}{(R_E + h)\cos\beta}\right] \tag{6.5}$$



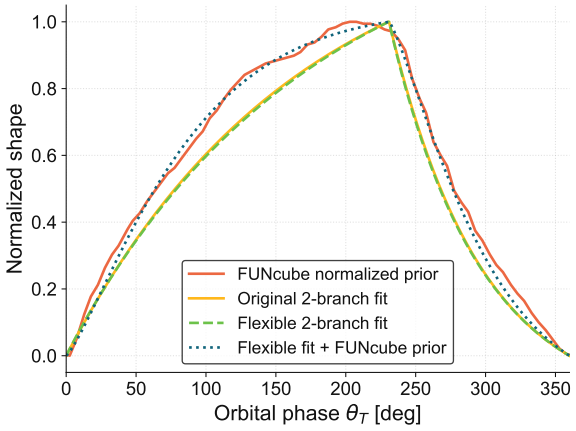
(a) Raw Delfi telemetry scatter with the phase-binned median and interquartile range.



(b) Normalized comparison between the Delfi median curve and the corresponding FUNcube reference curve.



(c) Delfi phase-binned median with the three fitted branch-model variants in temperature units.



(d) Normalized comparison of the fitted branch-model variants against the Delfi median curve.

Figure 6.2: Phase-binned thermal curve construction and fit comparison for the Panel Xm channel.

This eclipse fraction is then converted into the eclipse-transition angle used by the orbit-template model:

$$\theta_E(t_{\text{days}}) = 360^\circ \left(1 - f_E(t_{\text{days}})\right). \tag{6.6}$$

This allows the sunlight-eclipse transition in the orbit template to vary slowly over the year. As illustrated in Figure 6.3a, the eclipse-geometry term adjusts the duration of the heating and cooling branches without arbitrarily changing the full orbit-template shape.

6.5.2. Annual Irradiance Modulation

In the clean benchmark, annual irradiance variation is represented by a bounded node-wise vertical offset. The driver is derived from a normalized relative irradiance term and scaled by node-specific amplitudes, with larger modulation assigned to external panels and smaller modulation assigned to more thermally buffered internal nodes. This allows the nominal temperature baseline to drift gradually over the synthetic year without being interpreted as a fault.

The Earth-Sun distance in astronomical units is approximated as

$$r_\odot(t_{\text{days}}) = 1 - e_\oplus \cos\left(\frac{2\pi(t_{\text{days}} - t_p)}{365}\right) \tag{6.7}$$

where $e_\oplus$ is the orbital eccentricity of Earth and t_p is the approximate day of perihelion. The relative

solar flux is then computed as

$$S_{\text{rel}}(t_{\text{days}}) = \frac{1}{r_{\odot}(t_{\text{days}})^2} \tag{6.8}$$

To use this as a bounded annual forcing term, the relative flux is normalized according to

$$S_{\text{norm}}(t_{\text{days}}) = \frac{S_{\text{rel}}(t_{\text{days}}) - \overline{S}_{\text{rel}}}{\max_t \left| S_{\text{rel}}(t_{\text{days}}) - \overline{S}_{\text{rel}} \right|} \tag{6.9}$$

The annual temperature contribution for channel j is then modelled as

$$\Delta T_{\text{ann},j}(t) = A_{\text{ann},j} S_{\text{norm}}(t) \tag{6.10}$$

where $A_{\text{ann},j}$ is the channel-specific annual modulation amplitude. The resulting annual offset is shown in Figure 6.3b, where the full orbit-scale waveform is shifted gradually over the year.

6.5.3. Spin-Induced Ripple and Nominal Noise

In the clean benchmark, spin-imprinted thermal ripple is represented by a small sinusoidal perturbation added to the orbit-scale template. The driver introduces controlled fine-scale nominal structure that is not fully captured by the compact heating and cooling template. This makes the later residual-detection problem more realistic, because the detector must tolerate small systematic variation rather than operate against an unrealistically smooth baseline.

The spin phase can be computed from the spin rate as

$$\phi_{\text{spin}}(t) = \int_0^t \omega_{\text{spin}}(\tau)\, d\tau = \omega_{\text{spin}} t \;\text{ for constant } \omega_{\text{spin}} \tag{6.11}$$

The spin-induced temperature ripple for channel j is then represented as

$$\Delta T_{\text{spin},j}(t) = A_{\text{spin},j} \sin\left(\phi_{\text{spin}}(t) + \phi_j\right) \tag{6.12}$$

where $A_{\text{spin},j}$ is the node-specific ripple amplitude and ϕ_j is the node-specific phase offset. The short-timescale ripple is shown locally in Figure 6.3c, where it appears as a small periodic perturbation superimposed on the smooth orbit template.

In the clean benchmark, low-amplitude nominal noise is added as part of the fault-free reference signal. This represents ordinary measurement and modelling variation in the nominal baseline. It is distinct from the ordinary-noise nuisance injections introduced later in Chapter 7: here, noise is part of nominal behaviour, while later nuisance injections are explicit observation-layer stress cases.

$$\epsilon_j(t) \sim \mathcal{N}\left(0, {\sigma_{j,\text{nom}}}^2\right), \tag{6.13}$$

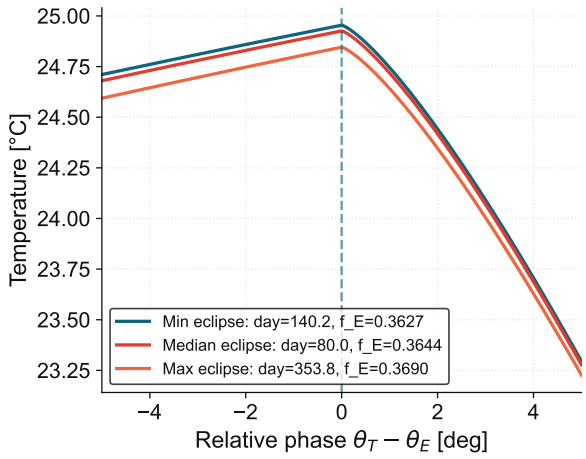
where $\sigma_{j,\text{nom}}$ is the nominal noise standard deviation for channel j.

6.5.4. Final Node-Wise Signal Composition

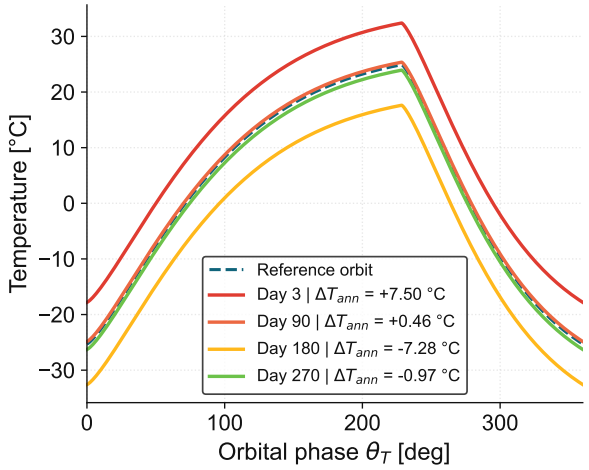
The final clean temperature signal for each node is composed from the fitted orbit-scale template, the seasonal eclipse-geometry modulation, the annual irradiance term, the spin-induced ripple, and nominal noise. This composition defines the fault-free thermal truth used by the synthetic benchmark. For each thermal node j, the clean nominal temperature is constructed as

$$T_{\text{clean},j}(t) = T_{\text{orbit},j}\left(\theta_T(t), \theta_E(t)\right) + \Delta T_{\text{ann},j}(t) + \Delta T_{\text{spin},j}(t) + \epsilon_j(t) \tag{6.14}$$

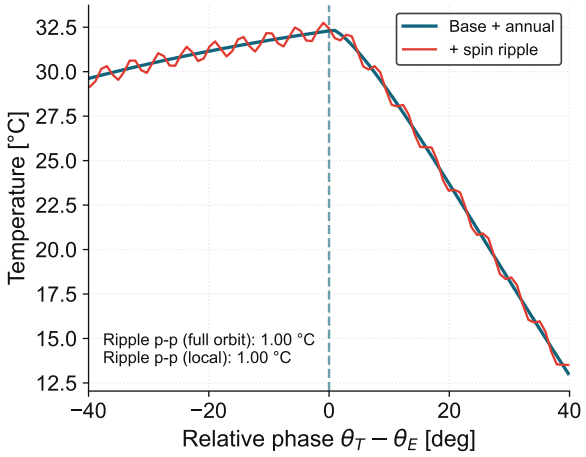
Here, $T_{\text{orbit},j}$ is the fitted orbit-scale thermal template evaluated sample by sample using the true-anomaly phase θ_T and eclipse-transition angle θ_E. The remaining terms represent annual irradiance modulation $\Delta T_{\text{ann},j}$, spin-induced ripple $\Delta T_{\text{spin},j}$, and low-amplitude nominal Gaussian noise ϵ_j.



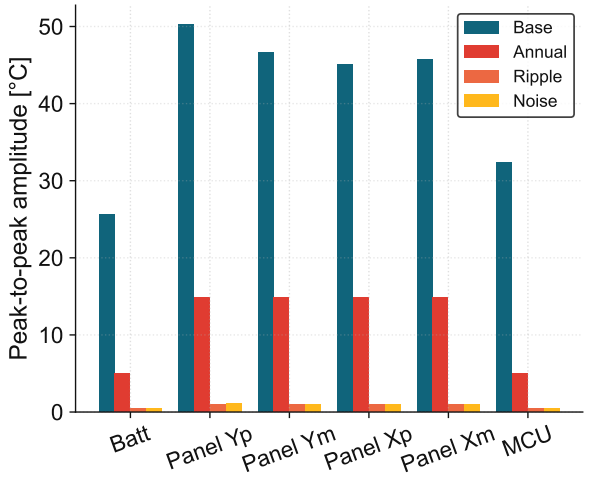
(a) Local effect of bounded eclipse-fraction variation, shown within ±30° of the eclipse transition.



(b) Annual irradiance modulation as an orbit-wise temperature offset.



(c) Local zoom of the spin/nutation ripple superimposed on the smooth nominal orbit.



(d) Node-wise peak-to-peak amplitudes of the nominal synthesis components.

Figure 6.3: Clean nominal reference telemetry construction from orbit-scale templates and nominal environmental drivers.

This composition provides a clean, continuous, fault-free thermal reference. It captures the main behaviours that the detector should treat as nominal: orbit-scale heating and cooling, slow seasonal modulation, annual baseline drift, spin-imprinted ripple, and ordinary low-level noise. The relative peak-to-peak amplitudes of these nominal components are summarized in Figure 6.3d.

The node-wise amplitudes and noise settings in Table 6.4 are benchmark parameters chosen to produce order-of-magnitude plausible nominal variability for detector development. They are not intended to predict exact Delfi Twin flight temperatures. The clean reference is therefore used to test detector behaviour under controlled conditions, not to certify a spacecraft thermal model.

The annual irradiance coefficients assign larger baseline modulation to external panels than to internal nodes. This is consistent with Delfi-C3 data [8] and with the expectation that directly exposed panels respond more strongly to annual solar-flux and illumination changes than the thermally buffered internal stack. The spin-ripple and Gaussian-noise settings are kept small relative to the orbit-scale waveform, so that they broaden the nominal residual distribution without creating fault-like disturbances. The selected values are summarized in Table 6.4.

Table 6.4: Node-wise benchmark parameters for annual irradiance modulation, spin-imprinted ripple, and low-amplitude nominal Gaussian noise.

Component / Parameter	Battery	Panel Y+	Panel Y-	Panel X+	Panel X-	MCU
Annual irradiance modulation Amplitude coefficient $A_{\text{ann},j}$ [°C]	2.5	7.5	7.5	7.5	7.5	2.5
Spin-induced ripple Ripple amplitude $A_{\text{spin},j}$ [°C]	0.25	0.50	0.50	0.50	0.50	0.25
Spin-induced ripple Phase offset ϕ_j [rad]	0	0	π	$\pi/2$	$3\pi/2$	0
Nominal Gaussian noise Standard deviation $\sigma_{j,\text{nom}}$ [°C]	0.05	0.10	0.10	0.10	0.10	0.05

6.6. Construction of the Clean Reference Dataset

The preceding sections described how representative orbit-scale templates were derived and augmented with seasonal eclipse-geometry modulation, annual irradiance modulation, spin-induced ripple, and nominal Gaussian noise. These components were combined to produce a long-duration clean thermal reference dataset for later synthetic injection and Anomaly Detection (AD) experiments.

The clean reference dataset is generated by evaluating the node-wise signal composition over the full synthetic mission duration. The final output is a 5 yr clean multichannel thermal truth dataset containing 6 temperature channels: the battery, MCU, and four solar panel temperature nodes. The exported dataset stores the generated temperatures together with the driver variables needed to reproduce and interpret them, including elapsed time, sample index, orbit index, orbital phase, eclipse fraction, eclipse-transition angle, beta angle, normalized solar flux, spin phase, and spin rate. The generation settings are summarized in Table 6.5.

Table 6.5: Generation-specific settings used to construct the clean nominal thermal truth dataset. The orbital environment parameters are defined separately in Table 6.2.

Setting	Value used
Duration	5 yr (365 d/yr)
Number of samples	10 512 000
Sampling cadence	15 s
Assumed LTAN	10.5 h
Spin rate	5 ° s^{-1}
Random seed	1
Export format	Parquet dataset

Table 6.6: Variable groups stored in the clean nominal thermal truth dataset.

Variable group	Examples
Time	Elapsed time, sample index, orbit index
Orbital	Orbital phase, eclipse fraction, eclipse-transition angle
Environmental	Beta angle, normalized solar flux, spin phase, spin rate
Temperature	Clean battery, solar panels, MCU

6.7. Assumptions and Implications for Operational Use

The clean reference dataset is a controlled nominal dataset for benchmark construction, not a mission-ready thermal simulator or flight prediction baseline. The following assumptions should be reconsidered before adapting the clean-reference approach for operational use:

- **Idealized orbital phase:** The phase variable $\theta_T(t)$ assumes regular orbit progression. In flight, it must be reconstructed from spacecraft time and orbit knowledge, such as Two-Line Element set (TLE) propagation, Global Navigation Satellite System (GNSS), or orbit determination.
- **Simplified eclipse geometry:** The eclipse-transition angle $\theta_E(t)$ is derived from an idealized eclipse fraction. Operationally, ingress and egress should be computed from Sun–Earth–spacecraft geometry or validated against solar-power telemetry.
- **Beta-angle approximation:** Seasonal eclipse variation uses a simplified β-angle model. Flight systems require beta angle from the actual orbital plane and Sun vector.

- **Sun geometry and declination:** Seasonal Sun geometry is simplified. Operational use requires ephemeris- or Sun-vector-based computation where eclipse timing matters.
- **Annual irradiance term:** The irradiance term models Earth–Sun distance variation, not measured solar-cell current. Solar-cell current may provide useful context, but only as an indirect proxy for illumination and power-generation conditions.
- **Fixed LTAN:** The clean reference assumes a fixed LTAN. In flight, LTAN drift and orbit differences affect illumination timing and eclipse duration.
- **Stable thermal template:** Orbit-scale temperature templates are assumed stable aside from the modelled nominal drivers. Real systems vary with attitude, mode changes, degradation, and subsystem activity.
- **Simplified attitude dynamics:** Spin-imprinted ripple is modelled as fixed-rate sinusoidal variation. Actual spacecraft may exhibit changing spin, nutation, pointing modes, or attitude-dependent heating.
- **Gaussian noise model:** Sensor noise is assumed Gaussian and low-amplitude. Flight data may include quantization effects, non-Gaussian noise, and temperature-dependent sensor behaviour.
- **No operational context:** The model excludes effects from ADCS, payload operation, radio transmission, battery cycling, heater control, and safe-mode behaviour.
- **No data-quality issues:** Contact gaps, missing samples, corruption, and sensor faults are excluded from the clean reference and introduced later during dataset generation.
- **Benchmark use only:** The dataset is suitable for controlled detector development and injection studies, but would require validation or adaptation before use as a flight prediction baseline.

6.8. Chapter Summary

This chapter constructed the clean nominal thermal truth dataset used as the substrate for the synthetic benchmark. The dataset combines fitted orbit-scale temperature templates with simplified seasonal eclipse-geometry modulation, annual irradiance variation, spin-imprinted ripple, and nominal noise. The result is a repeatable, fault-free, multichannel thermal reference at the telemetry cadence used throughout the thesis.

The main output of this chapter is the benchmark-construction method and generated clean reference dataset. The chapter does not claim to produce a flight-certified thermal simulator or a complete physical model of Delfi Twin. Instead, it provides a controlled nominal baseline into which Chapter 7 injects labelled telemetry pathologies and thermal faults. This separation allows later evaluation to distinguish nominal thermal variation, benign telemetry nuisances, telemetry-quality anomalies, and alert-worthy physical thermal faults.

7

Dataset Generation

This chapter constructs the labelled dataset used to develop, tune, and evaluate the on-board Anomaly Detection (AD) pipeline. Section 2.6 established why existing spacecraft telemetry datasets are valuable for realism but insufficient as direct quantitative ground truth for this thesis: they do not provide Delfi Twin-specific thermal events with known affected channel, onset, offset, anomaly pattern, and operational interpretation. Chapter 4 defined the relevant telemetry pathologies and thermal fault families, and Chapter 6 constructed the clean nominal thermal reference.

The role of this chapter is therefore implementation-focused. It turns the clean nominal reference into observed telemetry by injecting controlled benign telemetry nuisances, telemetry-quality anomalies, and mission-relevant physical thermal faults. The resulting benchmark is not intended to reproduce every possible spacecraft failure. Instead, it provides a controlled event-level evaluation substrate in which the nominal signal is known, injected events have explicit labels, and detector behaviour can be evaluated against reproducible ground truth.

The dataset supports four evaluation needs. First, it tests recovery of alert-worthy telemetry-quality anomalies and physical thermal faults. Second, it tests suppression of benign telemetry nuisances. Third, it supports evaluation of event-lifecycle logic, including event start, persistence, clearing, hysteresis, and gap handling. Fourth, it verifies that the detector output remains interpretable at the channel and event level.

Deployment-Oriented Workflow Steps

4. Design the dataset and ground-truth protocol
Specify the data features, event label semantics (onset/offset), scenario suite, and train/validation/test splits that ensure result significance and reflect operational conditions.

7.1. Benchmark Strategy Adopted for this Thesis

Section 2.6 compared dataset construction strategies and justified synthetic injection into a mission-relevant nominal baseline. This chapter implements that strategy for Delfi Twin temperature telemetry.

The synthetic event construction is informed by generic labelled telemetry-anomaly primitives, such as outliers, gaps, noise, shifts, trends, resolution changes, frequency changes, and amplification effects, as represented in the synthetic telemetry library of Schefels et al. [60]. In this thesis, these primitives provide a conceptual reference for the thermal AD problem rather than a direct software dependency. Generic observation-layer behaviours motivate the benign telemetry nuisances and telemetry-quality anomalies, while mission-specific thermal-fault families are defined separately to exercise residual-to-event detection under physically interpretable temperature deviations.

The starting point is the clean nominal temperature signal generated in Chapter 6. For each thermal node j, the clean signal $T_{\text{clean},j}(t)$ represents fault-free nominal temperature behaviour. Observed

telemetry is then generated by applying an injected deviation $\Delta T_{\text{inj},j}(t)$, giving

$$T_{j,t}^{\text{obs}}(t) = T_{\text{clean},j}(t) + \Delta T_{\text{inj},j}(t) \tag{7.1}$$

Depending on the scenario, the injected deviation may represent benign nuisance behaviour, a telemetry-quality anomaly, or a physical thermal fault. Missing-data events are handled differently: the observed stream is masked rather than shifted, so the injected effect represents temporary loss of observability rather than a thermal offset.

The design principle is to keep each injected event *interpretable*. Each event is assigned explicit metadata, including affected channel, onset, offset, subtype, anomaly pattern, direction, alert-worthiness, and injection parameters. This makes the benchmark suitable for event-level evaluation rather than only point-wise anomaly scoring. The detector can therefore be assessed on whether it forms meaningful bounded events, attributes them correctly, suppresses nuisance behaviour, and produces useful alert evidence.

Real spacecraft datasets remain important, but they are not used as direct event ground truth in this chapter. Their role is to inform plausible nominal behaviour, anomaly patterns, and telemetry realism. The labelled synthetic benchmark provides the controlled event truth needed for quantitative evaluation, while real telemetry is reserved for later qualitative stress testing and interpretation.

7.2. Implemented Event Families

Chapter 4 defined the broader set of telemetry pathologies and thermal fault families relevant to Delfi Twin temperature telemetry. This chapter implements the subset needed for the synthetic benchmark. The implemented event families were selected because they are observable in the chosen temperature channels, can be parameterized without a full thermal Failure Modes, Effects, and Criticality Analysis (FMECA), and provide the anomaly patterns needed to evaluate the residual-to-event pipeline.

The benchmark includes two observation-layer categories and three primary thermal fault categories. Observation-layer events test whether the detector can distinguish benign telemetry imperfections from telemetry-quality anomalies that corrupt interpretation of the signal. Thermal fault events test whether the detector can recover physically meaningful temperature deviations in the battery, Microcontroller Unit (MCU), and solar panel channels. The implemented event families are summarized in Table 7.1.

The labelled event benchmark focuses on event classes that can be generated with clear onset and offset semantics and evaluated directly against labelled ground truth. This makes the dataset suitable for testing event formation, persistence, hysteresis, gap handling, nuisance suppression, and detector-faithful alert explanations.

Several fault families identified in Section 4.7 are not implemented as primary labelled event injections. **SPIN** faults, representing attitude, spin, or illumination anomalies, are thermally observable but require coupled attitude, illumination, and thermal modelling to parameterize credibly. **DEG** and **COND** faults are also not injected as discrete labelled events because their expected behaviour is slower and more relevant to predictor-validity assessment than bounded event recovery. These deferred families are retained as robustness drivers and future-work targets rather than primary event-level labels.

7.3. Shared Event Representation and Label Schema

All injected events use a common event-level label schema, presented in Table 7.2. This thesis-specific schema allows benign telemetry nuisances, telemetry-quality anomalies, and thermal faults to be evaluated using the same event-level logic.

Each label records the affected channel, event timing, anomaly category, pattern type, alert-worthiness, and injection parameters. The `alert_worthy` field separates robustness stressors from events that should produce stable alerts. Benign nuisances are included primarily to test whether the detector suppresses minor telemetry disturbances. Telemetry-quality anomalies and thermal faults are generally treated as alert-worthy because they either compromise interpretation of the telemetry stream or represent physically meaningful thermal deviations.

Table 7.1: Event families and robustness drivers represented in the synthetic benchmark.

Family	Role	Purpose in this thesis	Representation
Benign telemetry nuisances	Robustness stressor	Represents ordinary data effects such as quantization, isolated spikes, low-amplitude noise, and missing data.	Labelled nuisance intervals.
Telemetry-quality anomalies	Alert-worthy telemetry event	Represents abnormal measurement-chain behaviour such as pinned sensor values, and abrupt calibration steps.	Labelled alert-worthy events.
HEAT: heater control failure	Injected thermal fault	Represents a heater stuck ON or OFF on the battery or nearby internal hardware.	Labelled thermal-fault events.
MODE: self-heating anomaly	Injected thermal fault	Represents unexpected excess or missing self-heating from a subsystem.	Labelled thermal-fault events.
RUN: battery over-temperature	Injected thermal fault	Represents a rapidly emerging battery over-temperature or incipient runaway signature.	Labelled thermal-fault events.
DEG / COND	Predictor-validity stressor	Represents slow thermal-baseline or coupling changes that can make a fixed predictor invalid.	Evaluated through specific robustness scenarios.
SPIN	Deferred fault family	Represents attitude, spin, or illumination anomalies with thermal consequences.	Not implemented in the synthetic benchmark.

Table 7.2: Shared event-label fields used for injected dataset events.

Field	Meaning
`event_id`	Unique event identifier.
`split`	Dataset split: training, validation, or evaluation.
`category`	Event category: benign nuisance, telemetry-quality anomaly, or thermal fault.
`subtype`	Specific injected event type, such as spike, pinned value, HEAT, MODE, or RUN.
`node`	Primary affected temperature channel.
`start_time`, `end_time`	Event onset and offset in dataset time.
`start_idx`, `end_idx`	Event onset and offset in sample indices.
`direction`	Positive, negative, mixed, or not applicable.
`anomaly_types`	Type I–V temperature-pattern classes.
`alert_worthy`	Whether the event should be recovered as a stable detector event.
`parameters`	Injection-specific values, such as amplitude, duration, ramp time, or saturation bound.

All event families are parameterized using the same basic dimensions: duration, magnitude, affected channel or channels, timing placement, and temporal evolution. This allows point-like artifacts, interval telemetry pathologies, and thermal faults to be generated within one consistent event-level framework. For example, a spike may affect one sample with a fixed amplitude, a telemetry gap may mask several minutes of data, and a battery over-temperature event may grow non-linearly over a short interval.

7.4. Observation-Layer Injections

Observation-layer injections modify the measured telemetry stream rather than the physical thermal state. They represent measurement artifacts, packet effects, calibration issues, or temporary loss of observability. As explored in Section 4.6, not all telemetry artifacts have the same operational meaning. Some are benign nuisances that perturb the measured signal but should not trigger a fault alert, whereas others indicate that the telemetry readout itself has become unreliable. This thesis therefore separates observation-layer injections into *benign telemetry nuisances* and *telemetry-quality anomalies*.

7.4.1. Benign Telemetry Nuisances

Benign telemetry nuisances are expected imperfections of the observed telemetry stream. They are included to test whether the detector avoids escalating small, isolated, or observability-related artifacts into stable anomaly events. These events may perturb or mask the measured signal, but they are not intended to represent alert-worthy events.

The representative examples shown in Figure 7.1 illustrate the three main nuisance classes retained in the synthetic dataset: small quantization, isolated spikes, and ordinary noise. Figure 7.2 shows the resulting loss of observability during a communication blackout. The nuisance types injected in this work, and their expected detector behaviour, are summarized in Table 7.3.

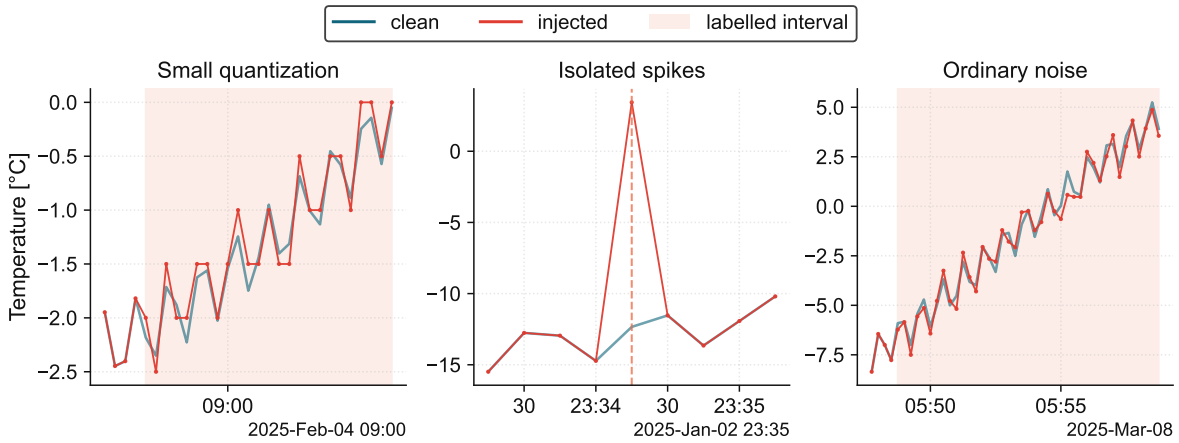


Figure 7.1: Representative benign telemetry nuisances injected into the synthetic dataset: small quantization, isolated spikes, and ordinary noise. These effects perturb the observed telemetry without representing alert-worthy events.

Small quantization is implemented by rounding reported temperature values to a fixed step size over a local window. Ordinary noise is modelled by adding zero-mean Gaussian noise,

$$\eta_t \sim \mathcal{N}(0, \sigma_{\text{noise}}{}^2) \tag{7.2}$$

where σ_{noise} controls the noise strength.

For isolated spikes, let $T_{\text{clean},t}$ be the clean nominal telemetry and $T_{\text{obs},t}$ the injected telemetry. In a selected local window $W = \{t_0, \dots, t_1\}$, spike amplitudes are scaled by

$$\delta_t = b_t \kappa_t s_W, \qquad s_W = \text{std}\left(T_{\text{clean},t} : t \in W\right) \tag{7.3}$$

Table 7.3: Benign telemetry nuisances injected into the synthetic dataset.

Subtype	Injection behaviour	Expected detector behaviour
Small quantization	Reported temperature values are rounded to a coarser step size over a local interval.	Should not trigger stable alert events.
Ordinary noise	Low-amplitude zero-mean Gaussian noise is added over a local interval.	May increase residual variance, but should not form stable alerts.
Isolated spikes	One or a few samples receive positive or negative impulse-like deviations.	Should be suppressed by persistence and event duration logic.
Contact gaps / missing data	Samples are masked as unavailable over local or recurring gap intervals.	Should terminate or reset evidence rather than bridge across missing telemetry.

where $b_t \in \{-1, +1\}$ sets the spike direction and κ_t is the spike factor. Noise and spike injections are applied directly to the clean signal: Gaussian perturbations are added across noise windows, while spike events superimpose a single-sample or short-duration offset δ_t on the nominal value.

Missing-data events are represented using a validity mask. Recurring contact gaps are placed using orbit-phase-based windows, reducing observability without implying any underlying thermal fault.

7.4.2. Telemetry-Quality Anomalies

Telemetry-quality anomalies represent abnormal behaviour of the sensing, readout, or processing chain. They do not necessarily indicate a physical thermal fault, but they can make the temperature stream unreliable or mimic a true thermal deviation. Unlike benign nuisances, they are treated as alert-worthy because they may require filtering, flagging, or operator review. The injected telemetry-quality anomaly types are summarized in Table 7.4, with representative examples shown in Figure 7.3.

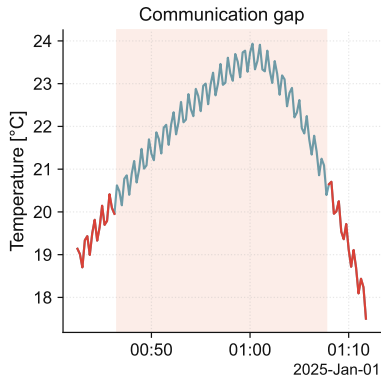


Figure 7.2: Representative communication gap injected into the synthetic dataset.

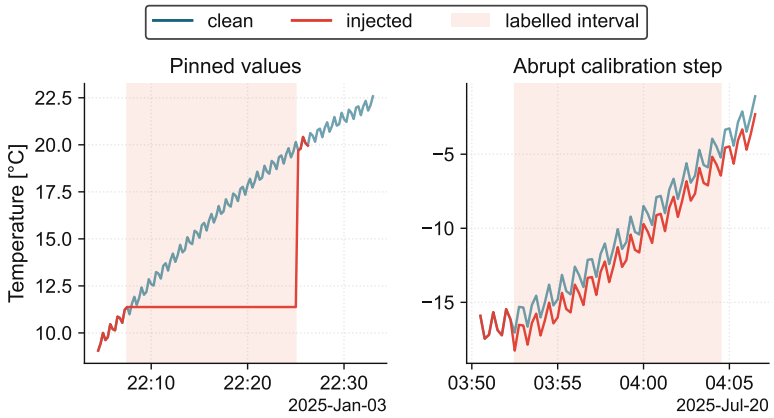


Figure 7.3: Representative telemetry-quality anomalies injected into the synthetic dataset.

Table 7.4: Telemetry-quality anomalies injected into the synthetic dataset.

Subtype	Injection behaviour	Expected detector behaviour
Pinned values	Temperature is held fixed at a constant value over an interval.	Should be detected as an abnormal flat segment rather than interpreted as a stable thermal regime.
Abrupt calibration step	A persistent positive or negative offset is added while preserving the local waveform shape.	Should be detected as a sustained level shift, but interpreted as telemetry-quality unless other evidence supports a physical fault interpretation.

7.5. Thermal Fault Injections

Thermal fault injections represent physically interpretable deviations in the underlying spacecraft thermal state and form the primary mission-relevant anomalies targeted by the AD pipeline. Unlike telemetry-quality anomalies, these events are intended to alter the expected thermal behaviour of the spacecraft itself rather than merely distort the readout. The representative examples shown in Figure 7.4 illustrate the three retained thermal fault classes: heater control failure, component self-heating anomaly, and battery thermal runaway.

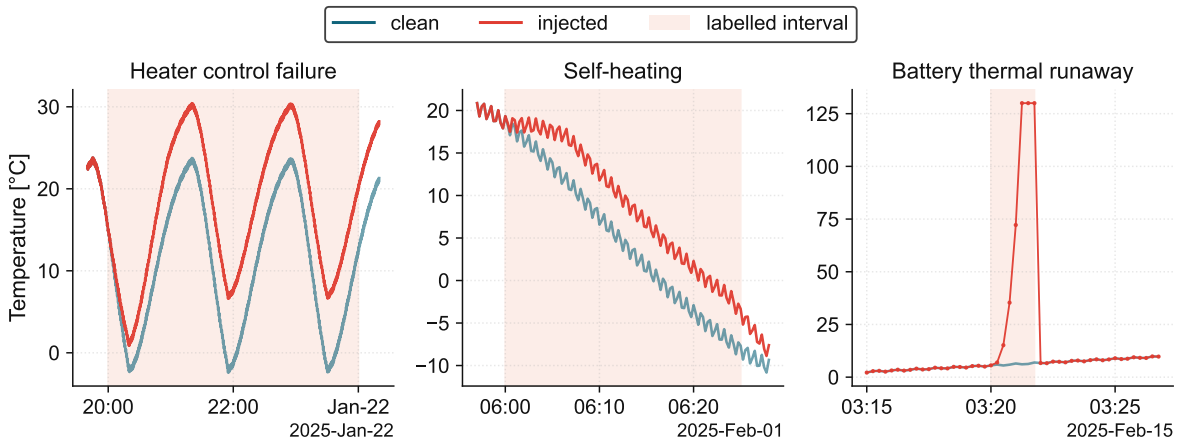


Figure 7.4: Representative mission-relevant thermal faults injected into the synthetic dataset: heater control failure, self-heating anomaly, and battery thermal runaway. These signatures represent physically interpretable deviations in the spacecraft thermal state.

7.5.1. Heater Control Failure

A heater control failure occurs when a component heater becomes stuck `ON` or `OFF`, producing sustained abnormal thermal regulation. In the temperature stream, a stuck `ON` heater appears as persistently hotter-than-nominal behaviour, while a stuck `OFF` heater appears as persistently cooler-than-nominal behaviour. The affected component may also show a modified orbit-scale amplitude, with weaker coupled effects in nearby channels.

In the synthetic dataset, this fault is implemented as a bounded multichannel thermal offset with smooth onset and offset. The battery channel is treated as the dominant affected channel and receives the largest deviation, while coupled channels, such as the MCU, receive scaled responses. The waveform amplitude is also modified to represent reduced swing for stuck `ON` behaviour and increased swing for stuck `OFF` behaviour. The event is labelled as an interval-type thermal fault on each affected channel.

7.5.2. Component Self-Heating Anomaly

A component self-heating anomaly occurs when a component or subsystem produces more or less internal heat than expected during an activity window. In the temperature stream, excess self-heating appears as locally hotter-than-nominal behaviour in the source component, with weaker coupled warming in nearby channels. Reduced or missing self-heating appears as colder-than-expected behaviour over the same activity interval.

In the synthetic dataset, this fault is implemented as a bounded multichannel thermal surplus or deficit. The dominant source channel, here the MCU, receives the largest positive or negative deviation, shaped by a ramp-in, plateau, and ramp-out profile. Coupled channels, such as the battery, receive scaled responses over the same window. The event is labelled as an interval-type thermal fault on each affected channel.

7.5.3. Battery Thermal Runaway

Battery thermal runaway represents a short, internally driven, accelerating battery over-temperature event. In the temperature stream, it appears as a rapid nonlinear increase in battery temperature that dominates local thermal behaviour and may approach the sensor upper limit. Nearby electronics may show weaker coupled increases, and telemetry loss may occur if the event affects subsystem operation.

In the synthetic dataset, this fault is implemented as a bounded convex ramp over a short interval, optionally followed by a saturated hold at the sensor maximum. The battery channel is treated as the dominant affected channel, while coupled channels, such as the MCU, receive smaller scaled ramps. Optional `NaN` masking represents telemetry loss after or during the event. The event is labelled as an interval-type thermal fault from onset through the active runaway phase on each affected channel; saturation or dropout may be included within the labelled anomaly window.

7.6. Rationale and Claim Boundary of Injected Event Families

The synthetic event families are selected to exercise the residual-to-event pipeline under controlled behaviours that are relevant for on-board temperature anomaly alerting. They are not intended to reproduce every possible Delfi Twin fault mode. Instead, the benchmark includes three complementary categories. Benign telemetry nuisances test whether the detector avoids stable alerts on small or short-lived perturbations. Telemetry-quality anomalies test whether the detector can flag measurements whose interpretation has become unreliable. Thermal-fault injections test whether physically meaningful deviations in spacecraft thermal state can be converted into bounded, explainable alert events.

Table 7.5 summarizes the rationale for each implemented event family and the main claim boundary associated with it. These boundaries define how the synthetic results should be interpreted: the benchmark supports controlled event-level evaluation, but it does not estimate real flight fault rates or exhaust all possible thermal behaviours.

Table 7.5: Rationale and claim boundaries of the injected synthetic event families.

Event family	Operational behaviour represented	Reason for inclusion in the benchmark	Main claim boundary
Small quantization	Low-amplitude discretization or measurement texture in nominal telemetry.	Verifies robustness against benign perturbations without triggering false alert events.	Simplified measurement effect; not a full sensor or ADC model.
Isolated spikes	Single-sample or short impulsive artifacts from corruption or packet disturbance.	Validates suppression of transient spikes to avoid misclassification as sustained thermal events.	Does not represent persistent physical heating or cooling.
Ordinary noise	Increased stochastic variation around nominal temperature.	Checks stability under benign noise and prevents unstable event formation.	Synthetic noise does not cover full spacecraft noise behaviour.
Communication gaps	Temporary loss of telemetry due to missing samples or coverage interruptions.	Evaluates gap-aware event handling and prevents false continuity across missing data.	Represents observability loss only, not underlying physical state.
Pinned values	Channel remains constant despite changing expected temperature.	Detects telemetry-quality degradation where measurements become unreliable.	Does not identify hardware cause of failure.
Abrupt calibration step	Sudden offset due to calibration or reference change.	Assesses response to sustained shifts in telemetry interpretation.	May resemble thermal offset; labelled here as telemetry-quality change.
Heater-control failure	Heater stuck `ON`/`OFF`, causing excess or missing heating and sustained thermal deviation.	Validates detection of sustained residuals and bounded event formation.	Not a detailed model of heater electronics or thermal coupling.

Continued on next page

Event family	Operational behaviour represented	Reason for inclusion in the benchmark	Main claim boundary
Component self-heating anomaly	Unexpected local heat generation from subsystem activity.	Validates detection of sustained positive residuals and interpretable thermal deviations.	Does not infer causal component state or operational mode.
Battery thermal runaway	Rapid or escalating positive temperature deviation in the battery channel.	Evaluates early-warning event formation under severe thermal growth.	Stress case only; not a validated electrochemical runaway model.

7.7. Deferred Fault Families and Robustness Drivers

Not all thermal fault families identified in Chapter 4 are injected as labelled events in the training, validation, and evaluation datasets. For quantitative event-level evaluation, the benchmark focuses on families that can be represented with clear onset, offset, affected-channel, and alert semantics, including telemetry-quality anomalies and the HEAT, MODE, and RUN thermal fault classes.

Three additional families, SPIN, DEG, and COND, are retained as deployment-relevant but are not implemented as primary labelled injections. Their signatures either require modelling beyond the benchmark scope or evolve too slowly to be treated as bounded events, making them more suitable as predictor-validity stressors than event-level anomalies. Their role is summarized in Table 7.6.

Table 7.6: Deferred thermal fault families and their role in the thesis.

Family	Reason not injected as labelled event	Role in this thesis
SPIN	Requires coupled attitude, illumination, and thermal modelling to produce a credible temperature signature.	Retained as future work and appendix-level analysis; motivates sensitivity to panel-amplitude, phase, and ripple mismatch.
DEG	Surface/coating degradation evolves slowly over mission life and does not naturally form a short bounded event.	Motivates slow thermal-bias and predictor-validity scenarios, especially for exposed panels.
COND	Thermal-interface degradation changes coupling, gradients, time constants, and orbit-envelope behaviour over long timescales.	Motivates predictor-mismatch scenarios involving node-specific offsets, amplitude changes, and thermal-envelope distortion.

SPIN represents attitude, spin, or illumination-driven thermal changes, expected to affect external panels through orbit-scale amplitude shifts, face-to-face gradients, or spin-ripple behaviour. A physically realistic injection would require coupled attitude and illumination modelling and is therefore excluded from labelled event generation.

DEG and COND are also not injected as labelled events. Surface and coating degradation can shift long-term thermal baselines, while thermal-interface degradation alters coupling and orbit-envelope behaviour. These effects are operationally important because they progressively invalidate a fixed nominal predictor; in this thesis, they instead motivate the predictor-mismatch and temperature-bias robustness scenarios used in evaluation.

Further physical motivation and parametrization details are provided in Appendix A. This separation keeps the benchmark focused on reproducible event-level evaluation, while deferred families remain part of the broader deployment discussion and future validation pathway.

7.8. Dataset Assembly & Splits

The final benchmark is assembled by applying the injection layers defined above to the clean nominal thermal truth from Chapter 6. The clean channels are preserved as the reference trajectory, while injected deviations are applied only to the observed telemetry streams. This separation keeps the nominal truth available for residual calculation, visualization, and diagnostic comparison.

The generation workflow is shown in Figure 7.5. First, the clean orbit-aware thermal baseline is copied into an observed telemetry representation. Observation-layer effects are then applied, including contact gaps, benign nuisances, and telemetry-quality anomalies. Mission-relevant thermal faults are injected into selected scenarios, modifying the observed stream and appending entries to the canonical label table. For multichannel events, separate label rows are stored per affected channel to distinguish dominant and coupled responses.

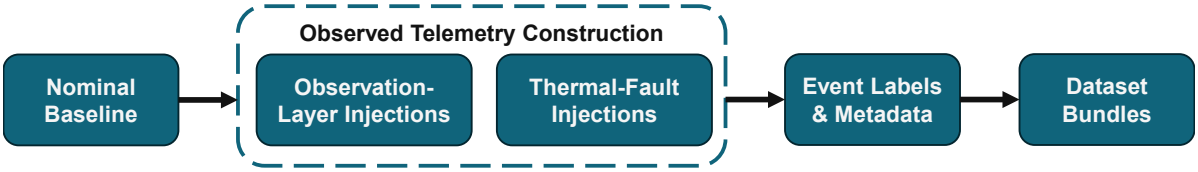


Figure 7.5: Synthetic telemetry generation workflow used to create labelled train, validation, and evaluation bundles.

To support model development and held-out evaluation under long-duration operational variability, the 5 yr thermal truth dataset is partitioned chronologically into a 1 yr training period, a 2 yr validation period, and a 2 yr evaluation period. The training set contains nominal thermal behaviour and is used to fit the expected-temperature predictor and estimate nominal residual statistics. The validation set includes labelled benign nuisances, telemetry-quality anomalies, and selected fault injections for tuning the residual decision logic. The evaluation set is held out until all predictor and detector parameters are fixed.

This separation is important because the predictor and decision logic require different evidence. The predictor must learn nominal seasonal and orbit-dependent structure, while the decision layer must be tuned against labelled deviations, persistence behaviour, hysteresis, and cumulative evidence. Keeping these roles separate prevents the final evaluation scenarios from influencing either the expected-temperature model or the alerting thresholds. The scenario content of each split is summarized in Table 7.7.

Table 7.7: Scenario content of the generated dataset bundles.

Split	Nominal	Benign nuisances	Telemetry-quality anomalies	Thermal faults	Mixed/overlap scenarios
Training	Yes	No	No	No	No
Validation	Yes	Yes	Yes	Selected	No
Evaluation	Yes	Yes	Yes	Yes	Future Work

"Selected" indicates scenario-specific events included for tuning or stress-testing.
Mixed/overlap scenarios assess thermal faults in the presence of observation-layer effects, rather than alone.

The workflow was implemented as a modular Python library with separate components for data loading, label handling, observation-layer injectors, thermal-fault injectors, campaign generation, plotting, and dataset export. The principal injection settings are summarized in Table 7.8.

Observation-layer effects are generated as campaigns with fixed densities per channel, whereas thermal faults are injected as scenario-specific events. This approach reflects their different roles: the former evaluate robustness to telemetry imperfections, while the latter define alert-worthy behaviours targeted by the detector. All event parameters are stored as metadata alongside labels, enabling full interpretability and deterministic regeneration of the benchmark, including randomized timing, duration, and severity.

The final output is a set of dataset bundles rather than a single time series. Each bundle contains the clean nominal telemetry, injected observed telemetry, channel-level event labels, and injection metadata. The clean telemetry provides the reference for residual analysis, the observed telemetry is presented to the detector, and the labels provide event-level ground truth. The resulting labelled event counts are shown in Table 7.9.

Table 7.8: Principal parameter settings used for the synthetic observation-layer injections and thermal fault injections.

Cat.	Injection	Placement / duration / density	Main parameters	Label
BN	Small quant.	RLW; 6 windows per channel; 40–160 samples (10–40 min)	• Quantization step $q \in \{0.1,\ 0.2\}$ °C; • Buffer = 20 samples	Int.
BN	Ordinary noise	RLW; 8 windows per channel; 60–240 samples (15–60 min)	• Noise $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$, strength $\sigma \in [0.03,\ 0.12]$ °C; • Buffer = 20 samples	Int.
BN	Isolated spikes	RLW; 8 windows per channel; 20–120 samples (5–30 min); 1–4 spikes per window	• Spike factor $\alpha \in [0.25,\ 0.8]$; • Amplitude scaled by local variability; • Buffer = 10 samples	Pt.
BN	Contact gaps	OPM; once per orbit; mean width $\approx 80^{\circ}$ true anomaly (≈ 21 min)	• Gap start/end from empirical phase stats; • 5-sample edge guard; • `NaN` masking across observed channels	Int.
TQA	Pinned values	RLW; 3 windows per channel; 20–100 samples (5–25 min)	• Hold mode $\in \{\texttt{hold\_first}, \texttt{hold\_last\_before}\}$; • Buffer = 20 samples	Int.
TQA	Abrupt calibration step	RLW; 2 windows per channel; 40–240 samples (10–60 min)	• Constant offset $\Delta T_{\text{step}} \in \pm[0.4,\ 1.8]$ °C; • Buffer = 20 samples	Int.
TF	Heater control failure	Scenario-defined fault interval; Figure 7.4: 1140 samples (≈ 4.75 h, $\sim$3 orbits)	• D: Battery, $w_{\text{batt}} = 1.00$; C: MCU, $w_{\text{MCU}} = 0.35$; • $\Delta T_{\text{batt}} \in [5,\ 8]$ °C (up to 10 °C); • Amplitude factor ON[0.82–0.93], OFF[1.08–1.18]; • Ramp-in 0.20–0.25; optional ramp-out 0–0.10	Int.
TF	Self-heating anomaly	Scenario-defined activity window; Figure 7.4: 120 samples (30 min)	• D: MCU, $w_{\text{MCU}} = 1.00$; C: Battery, $w_{\text{batt}} = 0.35$; • $\Delta T_{\text{src}} \in [5,\ 8]$ °C (up to 10 °C); • Ramp-in 0.20; ramp-out 0.20; plateau implicit	Int.
TF	Battery thermal runaway	Short scenario-defined event; Figure 7.4: 6 rise samples (90 s); optional 2-sample hold (30 s)	• D: Battery, $w_{\text{batt}} = 1.00$; C: MCU, $w_{\text{MCU}} = 0.20$; • MCU delay = 2 samples; • Convex ramp to sensor maximum $T_{\text{max}} = 130$ °C, exponent = 2.8; • Optional saturated hold and `NaN` masking	Int.

Cat.: BN = benign nuisance, TQA = telemetry-quality anomaly, TF = thermal fault. **Placement**: RLW = random local window, OPM = orbit-phase mask. **Label**: Int. = interval-type, Pt. = point-type. Thermal faults are scenario-specific rather than fixed-density yearly injections.

7.9. Evaluation Use of the Generated Dataset Bundles

The generated dataset bundles support two evaluation uses. First, the labelled validation and evaluation splits test whether telemetry-quality anomalies and thermal faults are recovered as bounded detector events. Second, predictor-mismatch scenarios test whether the residual-based detector remains operationally meaningful when the observed telemetry no longer matches the fixed nominal predictor. High recall under a well-aligned predictor therefore establishes controlled event recovery, but not robustness to later thermal-baseline drift.

The primary labelled-event evaluation uses the label schema in Table 7.2. A detector event is counted as detecting a labelled event when it overlaps the labelled interval on the corresponding node, allowing a small timing tolerance on the order of a single sample period. Telemetry-quality anomalies and thermal faults are treated as alert-worthy events. Benign nuisances are no-alert-expected robustness cases. Contact gaps are retained for observability and reset-handling tests, but are not counted as positive thermal or telemetry-quality detections.

Predictor mismatch is not treated as an injected event family in the labelled benchmark. Instead, mismatch scenarios are applied later as validation-stage stress tests. They shift the observed temperature stream before residuals are recomputed, while keeping the predictor and detector configuration fixed. Their purpose is to test whether the residual detector remains useful when the expected-temperature model is biased or misaligned with observed telemetry. The corresponding evaluation protocol and metrics are defined in Section 12.3, and the mismatch scenarios are summarized in Section 12.8.

Table 7.9: Number of injected labelled events in the validation and evaluation bundles. The training split is not included because it contains only nominal telemetry and no injected anomaly events.

Event type	Valid.	Eval.
Benign Nuisances	**108**	**207**
Isolated Spikes	66	123
Ordinary Noise	24	48
Small Quantization	18	36
Telemetry-Quality Anomalies	**12**	**30**
Pinned Values	6	18
Abrupt Calibration Step	6	12
Thermal Faults	**6**	**6**
Heater Control Failure	2	2
Component Self-Heating Fault	2	2
Battery Thermal Runaway	2	2
Total labelled events	**126**	**243**

7.10. Chapter Summary

This chapter implemented the labelled synthetic benchmark used to develop and evaluate the residual-to-alert AD pipeline. The benchmark is built by injecting controlled benign telemetry nuisances, telemetry-quality anomalies, and thermal-fault signatures into the clean nominal thermal reference generated in Chapter 6.

The implemented event families include benign telemetry nuisances, telemetry-quality anomalies, and three injected thermal-fault classes: HEAT, MODE, and RUN. Each event uses a shared event-level label schema that records the affected channel, event timing, anomaly category, pattern type, alert-worthiness, and injection parameters. SPIN, DEG, and COND are retained as deployment-relevant thermal fault families, but are not used as primary labelled event injections because they require additional modelling or mainly affect long-term predictor validity rather than short bounded events.

The generated bundles support both labelled event-level evaluation and later predictor-mismatch stress testing. The labelled validation and evaluation splits provide controlled event truth for telemetry-quality anomalies and thermal faults. Predictor mismatch is kept separate from the event-label taxonomy because it tests whether the detector remains meaningful when the expected-temperature model is biased or misaligned with observed telemetry, rather than representing an additional physical fault label. Chapter 8 uses this benchmark, together with the Part I requirements, to screen candidate AD pipeline architectures before the selected residual-to-alert design is implemented and evaluated.

8

Candidate Pipeline Architectures and Down-Selection

The preceding chapters established the deployment problem for the Delfi Twin prototype: on-board temperature Anomaly Detection (AD) must operate under embedded constraints, use telemetry with limited fault-level ground truth, convert evidence into bounded events, and provide compact, explainable alerts. Chapter 7 then constructed the labelled benchmark used to evaluate candidate pipeline behaviour.

This chapter applies that foundation to select the end-to-end AD architecture used in the remainder of the thesis. The goal is not to rank AD methods in general, but to determine which pipeline family is most appropriate for this specific deployment problem: temperature telemetry, limited ground truth, event-level alerting, operator explainability, and Microcontroller Unit (MCU)-class implementation.

The selected architecture must produce meaningful anomaly evidence, convert that evidence into bounded events, and remain feasible for real-time on-board execution. This chapter therefore first defines the architecture-selection criteria, then applies telemetry-driven and application-driven screening to the candidate method families reviewed in Section 2.8. It then derives the score-to-event decision requirements and selects the residual-to-alert architecture implemented in Chapters 9 and 10.

Deployment-Oriented Workflow Steps

5. Screen and down-select the end-to-end pipeline
Evaluate common approaches based on the characteristics of the telemetry and specific application, dismissing incompatible techniques.

8.1. Selection Criteria for the End-to-End Pipeline

Section 2.9 justified why a deployable detector must be considered as an end-to-end pipeline, not as a scoring model alone. A score is only an intermediate quantity: by itself, it does not specify whether an alert should be sent, how long an event lasted, which channel was affected, whether the behaviour persisted through noise or gaps, or what evidence should be included in the downlinked alert. The selected architecture must therefore define how telemetry is represented, how anomaly evidence is produced, how point-wise evidence becomes bounded events, and what information is transmitted to operators.

The selection considerations in Table 8.1 translate the earlier thesis foundation into pipeline-selection criteria. They combine telemetry properties, anomaly-label limitations, event-level alerting requirements, and embedded deployment constraints. These considerations are used to judge suitability for the Delfi Twin case study, not to rank AD methods in general.

Table 8.1: Architecture-selection considerations used to guide anomaly-scoring and pipeline selection.

ID	Consideration	Implication for Delfi Twin temperature telemetry
C1	Variable sequence coverage and duration	Telemetry may contain gaps, partial windows, or reset-related discontinuities, so methods should not require complete, equal-length sequences.
C2	Misalignment	Comparable thermal patterns may shift in time or orbital phase, so strict pointwise alignment can produce misleading scores.
C3	Unknown and variable anomaly length	Events may last from one sample to multiple orbits, so scoring should not depend on one fixed event duration.
C4	Similarity to normal data	Telemetry artifacts and operational variability can resemble thermal faults, so clean training data and explicit event labels are needed.
C5	Multi-regime nominal behaviour	Nominal temperature is not one fixed pattern; it varies with orbit phase, seasonal context, and operational state.
C6	Clustered or repeating anomaly-like patterns	Recurring gaps, operational transients, or repeated artifacts may form dense clusters and should not be treated as normal because they repeat.
C7	Dimensionality	A single temperature sample is low-dimensional, but window- or orbit-based representations can become high-dimensional and costly.
C8	Separability after projection	If windows are compressed into summary features, localized anomalies may be smoothed out or become indistinguishable from nominal data.
C9	Labels and class imbalance	True spacecraft fault labels are scarce and anomalies are rare, motivating synthetic injection with explicit event-level labels.
C10	Distance metric quality	Distance-based methods depend on whether the chosen metric captures meaningful similarity despite gaps, shifts, and phase changes.
C11	Train versus test compute	Training can occur offline, but inference must be causal and feasible within MCU-class timing, memory, and power constraints.
C12	”Anomalies are rare” assumption	Some anomaly-like behaviours may be frequent or operationally routine, so rarity alone is not a reliable definition of abnormality.

The considerations in Table 8.1 are not all used in the same way. Considerations related to variable coverage, misalignment, unknown event duration, multi-regime nominal behaviour, repeated anomaly-like patterns, dimensionality, embedded feasibility, and anomaly rarity directly inform the telemetry-driven screening matrix (C1–C3, C5–C7, C11, C12). Considerations related to similarity between anomalies and normal data motivate the clean nominal training split (C4) and explicit synthetic event labels (C9) developed in Chapters 6 and 7. Considerations related to compressed feature spaces and distance-based representations (C8, C10) motivate caution with methods that may smooth out localized thermal deviations or obscure operator interpretation.

Together, these considerations ensure that architecture selection is driven by the properties of the telemetry and deployment problem, rather than by generic benchmark performance alone.

8.2. Candidate Anomaly-Scoring Families Retained from Literature

Section 2.8 reviewed the main anomaly-scoring families used in spacecraft and time-series AD. For the deployment-oriented screening in this chapter, these families are carried forward as 6 candidate groups and assessed for suitability in a downlink-constrained thermal alerting pipeline, using Delfi Twin as the MCU-class case study:

F1 | Rules, Limits, Simple Statistics
Evaluates whether telemetry remains within a predefined nominal envelope (static or adaptive bounds), assigning higher scores to larger violations.

F2 | Forecasting/Prediction Residual
Compares observed values with a model's short-horizon predictions, treating larger prediction errors (residuals) as more anomalous.

F3 | Reconstruction
Compares the original signal with its reconstruction from a nominal representation, treating larger reconstruction errors as more anomalous.

F4 | Proximity / Distance
Evaluates similarity to nominal references under a chosen distance measure (nearest neighbours or prototypes), treating greater distance as more anomalous.

F5 | Distribution / Density
Evaluates how improbable an instance is under a nominal probability model (likelihood/density), treating lower probability (or lower density) as more anomalous.

F6 | Boundary and Ensembles
Evaluates whether an instance lies outside a learned normal region, or is consistently isolated by an ensemble, assigning higher scores to stronger outlierness.

The list is intentionally a compact reminder of the families reviewed in Chapter 2. The screening below focuses on how each family fits the telemetry, event-level alerting, explainability, and embedded-deployment requirements derived earlier.

8.3. Telemetry-Driven Screening

The first screening stage evaluates whether each scoring family is compatible with the data characteristics of Delfi Twin temperature telemetry. The screening questions are derived from the considerations in Table 8.1, and are outlined as follows:

Q1. Can the method handle variable coverage, gaps, or unequal data length? (C1)
Q2. Can the method tolerate phase shifts or misalignment of key features? (C2)
Q3. Can the method support unknown and variable anomaly duration? (C3)
Q4. Can the method support multi-regime nominal behaviour? (C5)
Q5. Is the method robust to repeating or bulk anomaly-like patterns? (C6)
Q6. Is the method robust when high-dimensional window representations are required? (C7)
Q7. Is the method suited for real-time, on-board embedded implementation? (C11)
Q8. Does the method avoid relying too strongly on the assumption that anomalies are rare? (C12)

For each question, the scoring families are rated using (+), (0), and (–). A **(+)** indicates that the family is generally favourable for the consideration, **(0)** indicates that suitability depends strongly on the implementation or data representation, and **(–)** indicates that the family is generally unfavourable. These ratings are qualitative design judgments used to support architecture selection, not performance guarantees. Scores were assigned by synthesizing documented assumptions and limitations from prior surveys and spacecraft-oriented benchmarking guidance [6, 10, 21, 35] with observations from the representative working examples developed in this thesis and documented in Section B.1. Full scoring rationales and supporting citations are provided in Appendix C.

Key Takeaways

The screening suggests that proximity/distance (F4) and distribution/density (F5) are the least suitable families for this mission context. They are sensitive to gaps, phase misalignment, feature representation, and repeating anomaly-like patterns. Boundary and ensembles (F6) remains plausible, but its suitability depends strongly on the chosen representation and model.

F1, F2, and F3 form the most defensible shortlist after telemetry-driven screening. F1 (rules, limits, simple statistics) is simple and robust but limited in early detection of subtle deviations. F2 (forecasting/prediction residual) directly compares observed temperature to expected behaviour, making it naturally suited to orbit- and season-conditioned thermal telemetry. F3 (reconstruction) is feasible but more representation-sensitive and requires greater configuration effort.

Table 8.2: Screening matrix for anomaly scoring families against telemetry-driven method-selection questions. Ratings use **+** (generally favourable), **0** (depends on instantiation/representation), and **–** (generally unfavourable). Conditional ratings (e.g., 0/–, 0/+) reflect strong dependence on model choice within the family.

Question		F1 *RLS*	F2 *FPR*	F3 *R*	F4 *PD*	F5 *DD*	F6 *BE*
Q1 *Coverage*	gaps or unequal length	+	0	0	–	0	0
Q2 *Alignment*	phase shifts or misalignment	+	0	0	–	–	0
Q3 *Duration*	variable anomaly length	+	+	+	0	0	0
Q4 *Regimes*	multi-regime nominal behaviour	+	0	0	+	+	0/–
Q5 *Frequency*	repeating or bulk patterns	0	0	0	–	–	0/–
Q6 *High-D*	high-dimensional windows	0	+	+	–	–	0/+
Q7 *Compute*	inference cost	+	0	0	–	0	0/+
Q8 *Rarity*	assumes anomalies are rare	+	+	+	0	–	0/–

Family key: RLS = rules/limits/simple statistics; FPR = forecasting/prediction residual; R = reconstruction; PD = proximity/distance; DD = distribution/density; BE = boundary/ensembles.

8.4. Application-Driven Screening

The second screening stage evaluates the candidate scoring families against application-level requirements for on-board thermal alerting. These requirements follow from the earlier discussion of score-to-event conversion in Section 2.9, detector-faithful alerting in Subsection 2.9.4, and embedded deployment constraints in Section 2.4. The screening focuses on whether each family can provide useful early evidence, support ground-operator interpretation, and remain practical to configure for the Delfi Twin case study.

In this chapter, these concerns are applied through three questions:

Q9. Can the family produce an anomaly score that becomes informative soon after fault onset, when deviations are still subtle, across (i) mean shifts, (ii) slow drift, and (iii) runaway-type trends?

Q10. Can the family provide a score-level explanation that can be understood and acted on by ground operators without specialized Machine Learning (ML) expertise?

Q11. Does achieving reliable, stable scoring require considerable tuning and expert intervention within this family across operating regimes (sensitivity to representation, hyperparameters/calibration)?

The resulting application-driven screening matrix is shown in Table 8.3. The same (+), (0), and (–) convention is used as in the telemetry-driven screening. Full scoring rationales, including the interpretation of each application-driven question, are provided in Appendix C.

Table 8.3: Application-driven screening matrix for anomaly scoring families. Ratings use **+** (generally favourable), **0** (depends on instantiation/representation), and **–** (generally unfavourable). Conditional ratings (e.g., 0/–, 0/+) reflect strong dependence on model choice within the family.

Question		F1 *RLS*	F2 *FPR*	F3 *R*	F4 *PD*	F5 *DD*	F6 *BE*
Q9 *Early warning*	mean shifts, slow drift, runaway	0	+	0	0	0	0/+
Q10 *Comprehensibility*	interpretability of explanations	+	+	0	0	0	0/–
Q11 *Complexity*	configuration, tuning, sensitivity	+	0	–	0	–	0/–

Family key: RLS = rules/limits/simple statistics; FPR = forecasting/prediction residual; R = reconstruction; PD = proximity/distance; DD = distribution/density; BE = boundary/ensembles.

Key Takeaways

Forecasting/prediction residual scoring (F2) is the strongest fit under the application-driven criteria. A prediction residual is naturally expressed in the measurement domain: observed temperature is hotter or colder than expected by a given amount. This makes the score easy to explain and directly relevant to early fault awareness.

Rules, limits, and simple statistics (F1) remain useful as interpretable reliability anchors, but they are less sensitive to subtle or gradual deviations unless additional derived features or context-conditioned thresholds are added. Reconstruction (F3) and boundary/ensemble methods (F6) remain plausible, but their higher configuration effort and less direct explanations make them less attractive for the selected on-board prototype pipeline.

Combining the telemetry-driven and application-driven screening, ***forecasting/prediction residual scoring is selected as the primary anomaly-scoring family***. Rules and limits are retained as simple threshold-based baseline comparators for later evaluation and as supporting elements within the residual-to-event decision logic. Reconstruction, proximity, density, and boundary/ensemble methods are not carried forward as the main pipeline family because they provide less direct measurement-domain evidence, require more representation or configuration choices, and are less naturally aligned with compact event-level thermal alerting on MCU-class hardware.

8.5. Derived Decision-Logic Requirements

Selecting a residual-scoring family does not by itself define an operational alert. A residual score indicates that observed temperature differs from expected temperature, but it does not define when an event starts, whether the evidence persisted, whether the event cleared, how missing telemetry should be handled, or what evidence should be included in a compact downlinked alert. The selected architecture must therefore include a residual-to-event decision layer.

Section 2.9 reviewed generic score-to-event primitives, including instantaneous thresholding, cumulative evidence, persistence, hysteresis, and gap handling. This section uses those primitives to derive the decision-layer requirements for the selected residual-scoring family. The detailed equations, state logic, parameters, and alert fields are defined later in Chapter 10.

For Delfi Twin temperature telemetry, the selected residual-to-alert pathway must satisfy 3 decision-layer requirements:

- **Detection rule:** distinguish expected from unexpected residual behaviour using instantaneous or accumulated evidence.
- **Event formation:** convert pointwise candidate evidence into stable bounded events using persistence, hysteresis, minimum-duration, merge/split, and gap-handling rules.
- **Alert reporting:** summarize event onset, duration, direction, severity, affected channel, and detector evidence in a compact operator-facing format.

The corresponding candidate decision-logic modules to fulfill this residual-to-event pathway are summarized in Table 8.4. The detailed formulation and final parametrization of this decision layer are developed in Chapter 10.

The instantaneous threshold provides fast response to large abrupt deviations such as spikes, calibration steps, or strong thermal excursions. Cumulative evidence is required because slow or weak deviations may remain below an instantaneous threshold for several samples while still representing a meaningful thermal change. Event lifecycle logic is required because candidate alarms are not yet operational alerts. Persistence, hysteresis, minimum-duration logic, gap termination, quiet reset, and transient-spike suppression determine whether detector output is stable and interpretable enough for operator use.

Table 8.4: Decision-logic modules required for residual-to-event conversion.

Module	Purpose	Role in this thesis
Instantaneous threshold	Detect large abrupt residual excursions.	Provides fast response to spikes, calibration steps, and strong thermal deviations.
Cumulative evidence	Accumulate weak directional residual evidence.	Supports early warning for sustained mean shifts, slow drift, and progressive faults.
Event lifecycle logic	Convert candidate alarms into stable alert events.	Provides persistence, hysteresis, minimum duration, gap handling, spike suppression, and clear logic.

8.5.1. Rationale for Combined Decision Logic

The anomaly patterns considered in this thesis place different demands on the residual-to-event pathway. Some require fast response to large residuals, while others require accumulation of weak evidence or careful lifecycle handling. Figure 8.1 illustrates representative cases that motivate the combined decision pathway.

A single large spike requires instantaneous sensitivity, but should not become a long alert. Repeated sub-threshold deviations and slow upward drift require cumulative evidence, because each individual residual may be too small to trigger an instantaneous threshold. Near-threshold noise motivates persistence and hysteresis to prevent alert chatter. Missing telemetry motivates gap handling, because cumulative evidence should not be carried blindly across observability gaps. Predictor mismatch motivates cautious event interpretation and explicit evidence reporting, since strong residual evidence may reflect model misalignment rather than a true physical fault.

The selected decision pathway therefore combines instantaneous threshold evidence, Cumulative Sum (CUSUM)-style cumulative evidence, and rule-based event lifecycle logic. This combination is lightweight enough for embedded implementation, interpretable enough for operator-facing alerts, and flexible enough to handle both abrupt telemetry artifacts and persistent thermal deviations.

Figure 8.1 should be interpreted as a set of thesis-generated design cases, not as literature benchmark results. Its purpose is to show why the selected residual pipeline requires instantaneous, cumulative, and lifecycle-based decision logic before the detailed formulation is defined in Chapter 10.

8.6. Selected Candidate Architecture

The preceding screening steps lead to a residual-based candidate architecture for Delfi Twin temperature telemetry. The architecture is summarized in Table 8.5.

Table 8.5: Selected candidate AD architecture and rationale.

Pipeline component	Selected approach	Rationale
Scoring family	Forecasting / prediction residual	Provides direct, interpretable evidence: observed temperature relative to expected nominal behaviour.
Nominal reference	Expected-temperature predictor	Avoids raw temperature thresholds by accounting for orbital and seasonal nominal variation.
Instantaneous evidence	Shewhart-like residual threshold	Provides a fast, lightweight response to large abrupt deviations.
Cumulative evidence	CUSUM-style directional accumulation	Detects small but persistent positive or negative residual shifts.
Event formation	Persistence, hysteresis, minimum duration, and clear logic	Converts pointwise candidate alarms into stable bounded events.
Robustness handling	Gap termination, quiet reset, and transient-spike suppression	Prevents stale evidence, gap bridging, and isolated artifacts from becoming misleading alerts.
Alert output	Detector-faithful event summary	Reports what deviated, in which direction, by how much, for how long, and which evidence triggered the event.

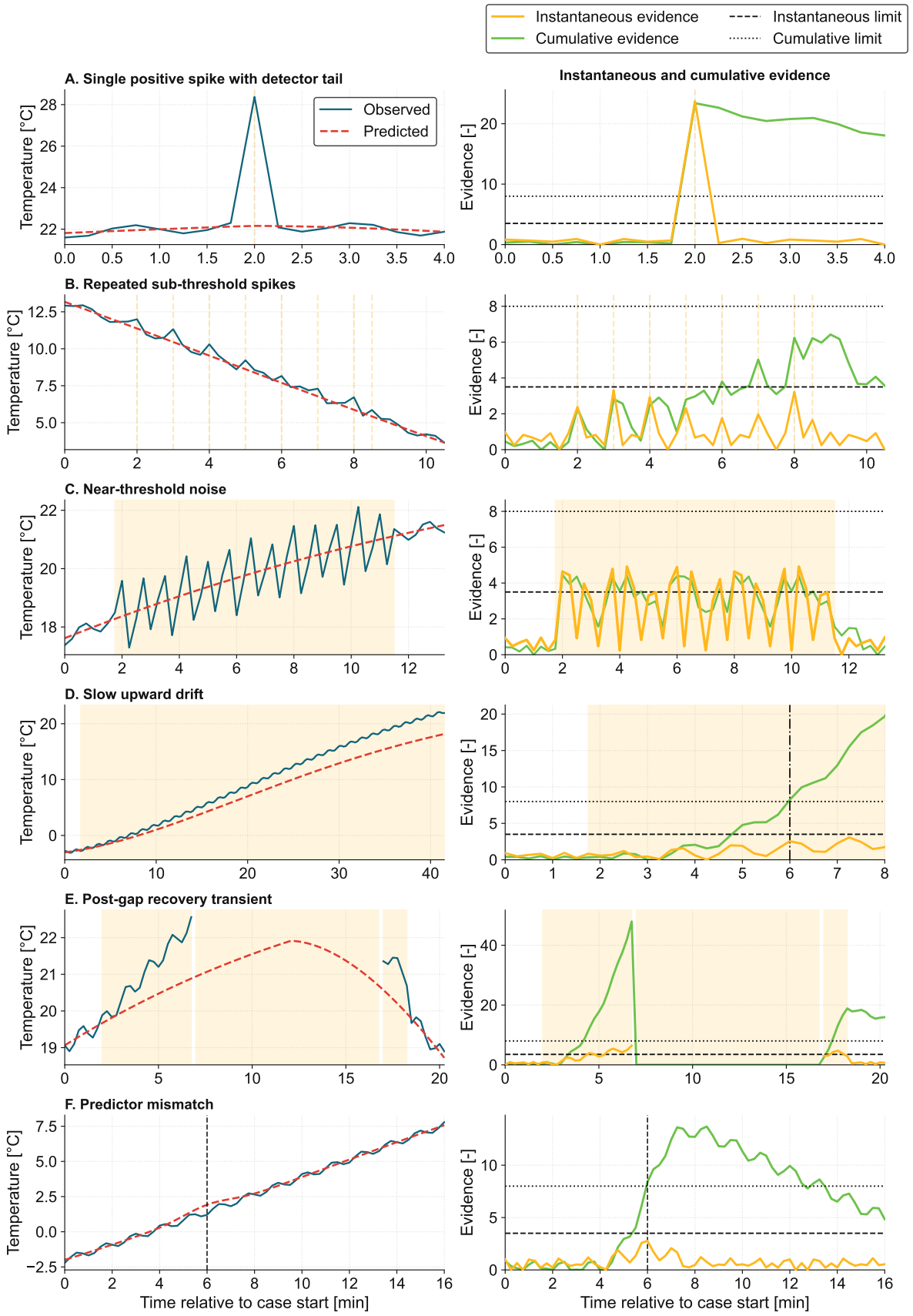


Figure 8.1: Representative behaviours motivating combined decision logic. Instantaneous evidence supports fast response, cumulative evidence supports persistent weak deviations, and event lifecycle rules suppress chatter, stale evidence, and gap-related artifacts.

The selected architecture uses expected-temperature prediction to generate measurement-domain residual evidence while accounting for orbit-scale and seasonal nominal thermal variation. Parallel instantaneous and cumulative evidence are then applied to the residual stream: instantaneous threshold evidence provides fast response to large abrupt deviations, while cumulative directional evidence supports detection of smaller but persistent residual shifts. Rule-based event formation stabilizes the alert stream through persistence, hysteresis, minimum duration, and clear logic, while robustness handling prevents stale evidence, gap bridging, and isolated artifacts from becoming misleading alerts. The resulting detector-faithful alert output reports the affected channel, deviation direction and magnitude, event timing, and trigger evidence.

The individual components are established primitives, but their combination and allocation into the Delfi Twin residual-to-alert pathway is the architecture selected in this thesis. The architecture is selected because it balances early sensitivity, measurement-domain explainability, and embedded feasibility. It does not attempt autonomous root-cause diagnosis. Instead, it produces detector-faithful evidence that can support ground-operator prioritization and review under constrained downlink.

8.7. Chapter Summary

This chapter translated the Part I literature review, Delfi Twin scope, and labelled benchmark requirements into the residual-to-alert AD architecture used in the remainder of the thesis. The down-selection considered telemetry realism, anomaly duration, phase dependence, multi-regime nominal behaviour, embedded feasibility, early warning, explainability, and configuration effort.

Forecasting or prediction residual scoring was selected as the primary scoring family because it provides a direct, measurement-domain indication of deviation from expected thermal behaviour. Rules and limits remain useful as lightweight decision primitives and are retained as simple threshold-based comparators for later evaluation, but they are not sufficient as the primary scoring approach. Reconstruction, proximity, density, and boundary or ensemble methods were not selected as the main architecture because they are more sensitive to representation choices, tuning effort, gaps, misalignment, rarity assumptions, or explanation complexity.

The selected architecture combines expected-temperature prediction, residual scoring, instantaneous threshold evidence, cumulative directional evidence, and rule-based event-lifecycle logic. This architecture is carried forward into the next two chapters: Chapter 9 defines the expected-temperature predictor, and Chapter 10 defines the residual decision logic, event lifecycle, and detector-faithful alert payload.

9

Thermal Prediction Models

Chapter 8 selected a residual-based anomaly-detection architecture for Delfi Twin temperature telemetry. In that architecture, the predictor provides the expected nominal temperature for each monitored node. The observed telemetry is then compared with this expected value to produce residual evidence for the decision layer in Chapter 10.

The predictor literature and candidate predictor families reviewed in Subsection 2.8.2 established the requirements for this component: the predictor should be lightweight, causal, interpretable, and accurate enough to produce useful residuals without suppressing anomaly-sensitive deviations. This chapter applies those requirements to the Delfi Twin residual detector by selecting, implementing, and verifying a compact expected-temperature predictor for the deployment-oriented pipeline developed in this thesis.

The objective is not general-purpose temperature forecasting. A predictor that minimizes forecast error is not necessarily the best predictor for Anomaly Detection (AD). For this thesis, the predictor must remove dominant nominal thermal structure while preserving anomaly-sensitive deviations. Underfitting leaves expected orbit-scale or seasonal variation in the residuals, increasing the risk of false alerts. Overfitting, or overly aggressive adaptation, can absorb early fault signatures that the detector is intended to score.

The fixed predictor selected in this chapter is a stretched-exponential orbit-template model with annual modulation. A bounded adaptive correction layer is also implemented as a robustness extension for predictor-mismatch experiments, but it is not treated as the baseline predictor or claimed to be flight-qualified. Its role is narrower: to test whether lightweight residual bias correction can reduce mismatch-driven false alerts while preserving local fault evidence.

Deployment-Oriented Workflow Steps

5. Screen and down-select the end-to-end pipeline
Evaluate common approaches based on the characteristics of the telemetry and specific application, dismissing incompatible techniques.

9.1. Predictor Role in the Residual Detector

The predictor is the expected-behaviour reference for the residual detector. For each monitored thermal node j, the predictor produces an expected nominal temperature

$$\widehat{T}_{\text{base},j}(t), \tag{9.1}$$

which is compared against the observed telemetry $T_{j,t}^{\text{obs}}(t)$ to form the residual

$$r_{j,t} = T_{j,t}^{\text{obs}}(t) - \widehat{T}_{\text{base},j}(t). \tag{9.2}$$

The downstream detector does not operate directly on raw temperature. It operates on normalized residuals, where the raw residual is adjusted using the nominal residual mean and spread for each node. Predictor quality is therefore judged by the residuals it leaves behind: nominal residuals should be centred near zero, bounded, and stable enough to support thresholding and cumulative evidence, while genuine deviations should remain visible for event detection.

This differs from a pure forecasting task. In forecasting, the main objective is usually to minimize prediction error for future samples. In residual-based AD, the predictor has a more constrained role: it should remove predictable nominal structure, such as orbit-scale heating and cooling and slow annual variation, while preserving deviations that may represent telemetry-quality anomalies or thermal faults. The residual stream should therefore remain sensitive to the event classes defined in Chapter 7 and should produce evidence that can be used by the decision logic developed in Chapter 10.

The predictor requirements used in this chapter follow from the literature review and scope analysis in Part I, provided in Table 9.1. These criteria combine general spacecraft telemetry-prediction concerns [71], with the specific requirements of residual-based AD [42, 34, 74, 7].

Table 9.1: Predictor-selection criteria for the residual-generation block.

ID	Criterion	Implication for this thesis	Source
C1	Physical plausibility	The expected-temperature profile should follow known orbit-scale spacecraft thermal behaviour, rather than only fitting historical data numerically.	[71]
C2	Non-stationary nominal behaviour	The predictor should remain useful as nominal temperature changes with orbit phase, eclipse transitions, operational context, and slow environmental drift.	[71]
C3	Cross-channel handling	Cross-channel dependence may improve realism, but is secondary here because interpretability and simple node-wise deployment are prioritized.	[71]
C4	Prediction stability	Predictions should remain stable over the horizon needed for residual generation, without rapid error growth or phase-dependent instability.	[71]
C5	Deployment suitability	Training may occur offline, but on-board inference should remain lightweight enough for causal Microcontroller Unit (MCU)-class use.	[71]
C6	Residual usefulness	The predictor should remove dominant nominal thermal structure without absorbing the deviations that the downstream detector is intended to score.	[42]
C7	Explainability	The predicted value should be explainable through interpretable drivers such as orbit phase, seasonal context, or bounded correction terms.	[34]
C8	Operational input requirements	The predictor should require only operationally realistic inputs. Methods needing detailed attitude, geometry, material, or unavailable context are less attractive.	[74]
C9	Implementation complexity	The method should be implementable, tunable, and verifiable within the thesis scope, without excessive preprocessing or architectural burden.	[7]
C10	Recalibration and bounded correction	The predictor should allow recalibration or bounded correction as nominal spacecraft behaviour changes over mission life.	[42]

9.2. Candidate Predictors Retained from Literature

Subsection 2.8.2 reviewed the relevant expected-behaviour and thermal-prediction model families. This chapter carries forward four candidates as design alternatives for the residual-generation block of the Delfi Twin pipeline: (i) a piecewise exponential orbit template with annual modulation, (ii) a stretched-exponential orbit template with annual modulation, (iii) an ARX/ARMAX-type statistical predictor with orbital context, and (iv) a stretched-exponential template augmented with lightweight adaptive correction. These candidates span the main predictor options considered plausible for a lightweight, interpretable residual detector.

As described during orbit-template fitting in Section 6.4, the piecewise exponential template is simple and physically interpretable, but its fixed exponential form is less flexible for fitting node-specific orbit-scale waveforms. The stretched-exponential template retains the physical structure of heating and cooling branches while allowing smoother, node-dependent transition shapes. The ARX/ARMAX candidate is lightweight and familiar as a time-series model, but it is less physically grounded for this use case and provides weaker operator interpretability. The adaptive-correction candidate is useful for robustness testing, but it is not selected as the baseline predictor because adaptation can reduce the local residual evidence that the detector is intended to score if it updates during anomaly onset.

9.3. Expected-Temperature Predictor Down-Selection

The preceding sections defined the predictor role, selection criteria, and candidate predictor families. This section applies those criteria to select the expected-temperature predictor used for residual generation in the Delfi Twin pipeline.

The shortlisted candidates are screened in Table 9.2. Ratings use **+** for generally favourable, **0** for mixed or conditional, and **–** for generally unfavourable. The ratings are thesis-specific design judgments for the Delfi Twin residual detector, not general rankings of thermal-prediction models.

Table 9.2: Screening matrix for shortlisted thermal predictor candidates against the forecasting method-selection questions. Ratings use **+** (generally favourable), **0** (mixed / conditional), and **–** (generally unfavourable).

Question		PEA *P1*	SEA *P1*	ARX *P3*	SEAC *P5*
Q1 *Physical plausibility*	nominal thermal trajectory	+	+	0	+
Q2 *Non-stationarity*	changing nominal behaviour	0	+	–	+
Q3 *Cross-channel*	exploit inter-channel dependence	–	–	0	0
Q4 *Forecast stability*	stability over required horizon	+	+	0	+
Q5 *Deployment*	lightweight / embedded suitability	+	+	+	+
Q6 *Residual usefulness*	preserves anomaly-sensitive residuals	+	+	–	0
Q7 *Explainability*	operator-meaningful prediction	+	+	+	+
Q8 *Input burden*	small, operationally realistic inputs	+	+	+	+
Q9 *Implementation*	thesis-scope complexity	+	+	0	0
Q10 *Updateability*	recalibration / lightweight correction	0	0	0	+

Candidate key: PEA = piecewise exponential template + annual term; SEA = stretched-exponential template + annual term; ARX = ARX / ARMAX-type statistical predictor; SEAC = stretched-exponential template + lightweight adaptive correction.

The screening favours the stretched-exponential template with annual modulation as the fixed predictor. It provides the best balance of physical plausibility, non-stationary nominal behaviour, prediction stability, residual usefulness, explainability, low input burden, and implementation simplicity. The piecewise exponential candidate is simpler, but less flexible for node-specific orbit-scale waveforms. The ARX/ARMAX candidate is lightweight, but less physically grounded for this use case and less directly interpretable as residual evidence. The adaptive-correction candidate is valuable for robustness testing, but it is not selected as the fixed baseline predictor. The same correction mechanism that recentres residuals under predictor mismatch could also reduce the local residual evidence needed to detect emerging telemetry-quality anomalies or thermal faults.

The selected baseline predictor is therefore the *fixed stretched-exponential orbit-template predictor with annual modulation*. The adaptive correction layer is retained as a robustness extension for predictor-mismatch evaluation, where its benefit can be assessed separately from the baseline residual detector.

9.4. Selected Fixed Predictor

Based on the qualitative screening and residual-generation requirements, the fixed predictor carried forward into the implemented residual detector is the stretched-exponential orbit-template model with

annual modulation. For each node j, the predictor is written as

$$\widehat{T}_{\text{base},j}(t) = f_{\text{stretch},j}(\theta_T(t), \theta_E(t)) + f_{\text{annual},j}(t). \tag{9.3}$$

where $f_{\text{stretch},j}$ represents the node-specific orbit-scale heating/cooling response and $f_{\text{annual},j}$ represents the selected annual modulation term.

The stretched-exponential template is not proposed here as a new general spacecraft thermal model. Its role is to provide a compact expected-temperature reference for the Delfi Twin residual-to-alert pipeline. The contribution lies in selecting, parameterizing, verifying, and embedding this predictor as part of the deployment-oriented AD architecture.

The model is intentionally compact. It captures the dominant nominal thermal structure, including orbit-scale heating and cooling and slow annual modulation, while leaving smaller spin-imprinted ripple and nominal noise in the residual stream. This is acceptable because the downstream detector is calibrated on the nominal residual distribution. The objective is not to remove every nominal fluctuation, but to produce residuals that are centred near zero, bounded, interpretable, and still sensitive to telemetry-quality anomalies and thermal faults.

Using the residual definition introduced in Section 9.1, ($\widehat{T}_{\text{base},j}$(t)) is the fixed stretched-exponential prediction with annual modulation. A positive residual means the observed node is warmer than expected, while a negative residual means it is cooler than expected. This sign convention is important because the decision layer in Chapter 10 separately tracks positive and negative cumulative evidence.

The adaptive correction is evaluated as an extension to this selected fixed predictor. It does not replace the fixed physical baseline. Instead, it adds a bounded correction term to test whether residuals can remain centred when the fixed predictor becomes mismatched to the observed telemetry.

9.5. Matched-Nominal Residual Consistency

The selected predictor is checked by the residuals it leaves under matched nominal conditions. This check evaluates whether the fixed predictor removes the dominant orbit-scale and annual thermal structure while leaving only the expected nominal residual variation used for downstream anomaly scoring. It should not be interpreted as operational predictor validation, because the nominal data are generated under the same matched assumptions used to define the benchmark.

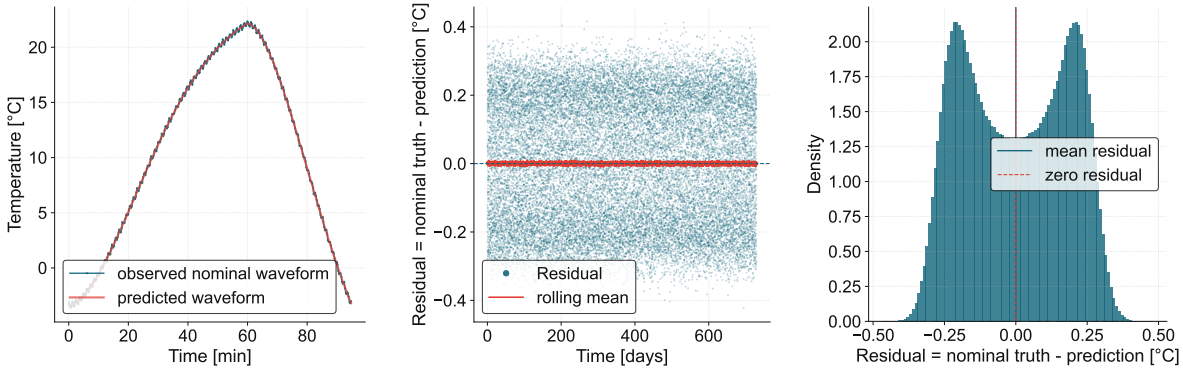
The residual diagnostics use Mean Absolute Error (MAE), Root-Mean-Square Error (RMSE), residual bias, residual standard deviation, and the 95th percentile absolute residual:

$$\begin{aligned} \text{MAE}_j &= \frac{1}{N}\sum_{t=1}^{N} |r_{j,t}|, & \text{RMSE}_j &= \sqrt{\frac{1}{N}\sum_{t=1}^{N} {r_{j,t}}^2}, \\ \text{bias}_j &= \frac{1}{N}\sum_{t=1}^{N} r_{j,t}, & \sigma_{r,j} &= \text{std}\big(r_{j,t}\big)\,. \end{aligned} \tag{9.4}$$

These metrics characterize whether the predictor leaves residuals that are centred, bounded, and suitable for normalization before the downstream AD decision layer. In this matched synthetic setting, the goal is not to drive the residuals to zero. The fixed predictor intentionally models the dominant orbit-scale and annual structure, while leaving small spin-imprinted ripple and low-amplitude nominal noise as residual texture for the detector to tolerate.

Figure 9.1 and Figure 9.2 show representative residual diagnostics for the battery and a solar-panel node. Each figure compares (a) the observed and predicted orbit-scale waveform, (b) the residual over the nominal validation period, and (c) the residual distribution. The internal nodes show lower residual scatter than the external panels, as expected from the larger deliberately unmodelled spin-induced ripple and higher nominal noise on exposed surfaces.

The expected nominal residual spread can be estimated from the components intentionally left outside the fixed predictor. Treating the unmodelled spin ripple as a sinusoid with amplitude $A_{\text{spin},j}$, its root-

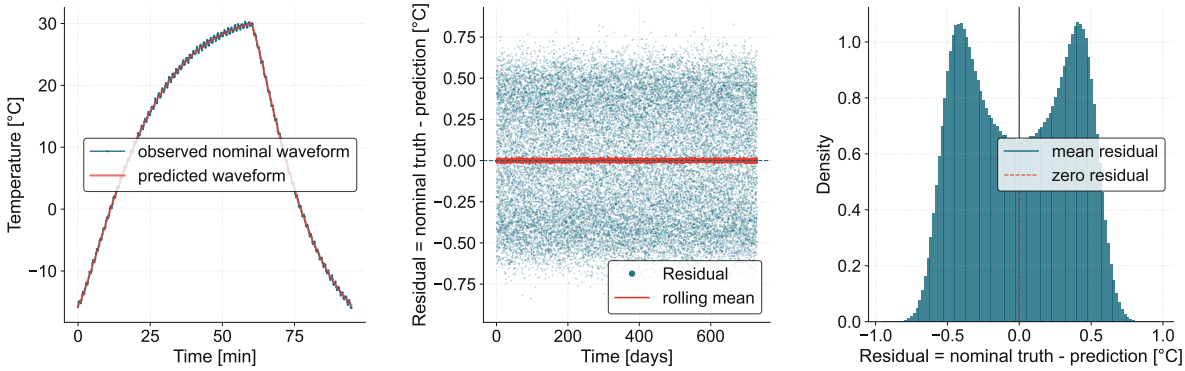


(a) Observed and predicted battery temperature over one representative orbit.

(b) Battery residual over the nominal validation period, with rolling mean.

(c) Distribution of battery residuals under nominal conditions.

Figure 9.1: Residual quality diagnostics for the selected fixed thermal predictor on the battery node.



(a) Observed and predicted Panel Xm temperature over one representative orbit.

(b) Panel Xm residual over the nominal validation period, with rolling mean.

(c) Distribution of Panel Xm residuals under nominal conditions.

Figure 9.2: Residual quality diagnostics for the selected fixed thermal predictor on the Panel Xm node.

mean-square contribution is $A_{\text{spin},j}/\sqrt{2}$. Combining this with zero-mean nominal noise gives

$$\text{RMSE}_{\text{expected}} \approx \sqrt{\left(\frac{A_{\text{spin},j}}{\sqrt{2}}\right)^2 + {\sigma_{\text{noise}}}^2} \tag{9.5}$$

For internal nodes, $A_{\text{spin}} = 0.25\,°\text{C}$ and $\sigma_{\text{noise}} = 0.05\,°\text{C}$, giving an expected nominal residual spread of approximately $0.184\,°\text{C}$. For external panels, $A_{\text{spin}} = 0.50\,°\text{C}$ and $\sigma_{\text{noise}} = 0.10\,°\text{C}$, giving approximately $0.367\,°\text{C}$. Table 9.3 compares these expected values with the observed residual metrics.

Table 9.3: Matched-nominal residual consistency of the selected fixed thermal predictor. The observed residual RMSE matches the expected contribution from deliberately unmodelled spin ripple and nominal noise.

Node group	Nodes	Spin A_{spin} [°C]	Noise σ [°C]	Expected RMSE [°C]	Observed MAE [°C]	Observed RMSE [°C]	P95$(\lvert r \rvert)$ [°C]
Internal nodes	Batt, MCU	0.25	0.05	0.184	0.162	0.184	0.295
External panels	Panel Yp, Panel Ym, Panel Xp, Panel Xm	0.50	0.10	0.367	0.325	0.367	0.590

The close agreement between the expected nominal residual spread and the observed RMSE indicates that, under matched synthetic nominal conditions, the residuals are dominated by deliberately unmodelled nominal texture rather than systematic predictor failure. The nominal residuals are also approximately centred, so their node-wise mean and spread can be used to define the scaler for normalized detector evidence.

This supports using the stretched-exponential orbit-template predictor as the fixed nominal reference for the implemented residual detector under matched benchmark assumptions. However, it should not be interpreted as evidence of operational robustness. The predictor is fixed after fitting, so any persistent mismatch between actual flight telemetry and the expected nominal model will appear as residual evidence. This motivates the predictor-mismatch robustness evaluation and adaptive-correction experiments later in Chapter 12.

9.6. Adaptive Predictor Extension for Mismatch Robustness

A fixed predictor can become misaligned when the observed thermal baseline no longer matches the fitted nominal model. This may occur through calibration offset, ageing, changes in thermo-optical properties, thermal-interface changes, or operating-context mismatch. For a residual detector, persistent predictor mismatch is important because it appears directly as residual bias and can drive detector evidence even when no new bounded thermal event has occurred.

Subsection 2.8.2 reviewed adaptive correction as a possible mitigation for predictor mismatch. Several correction types could in principle be considered, including additive bias correction, amplitude correction, phase or timing correction, or changes to the heating and cooling template shape. This thesis implements only the additive bias case. This is the lowest-complexity correction because it re-centres residuals without changing the orbit-template shape, transition timing, or thermal response parameters. Amplitude, phase, and shape adaptation would require stronger physical validation because they could more easily absorb thermal-fault evidence or mask changes in orbit-scale behaviour.

The corrected predictor for node j is therefore written as

$$\widehat{T}_{\mathrm{corr},j}(t) = \widehat{T}_{\mathrm{base},j}(t) + b_j(t). \tag{9.6}$$

where $b_j(t)$ is the applied bias correction to node j. Three correction structures are considered:

$$\text{common:} \quad \widehat{T}_{\mathrm{corr},j}(t) = \widehat{T}_{\mathrm{base},j}(t) + b_{\mathrm{common}}(t), \tag{9.7}$$

$$\text{node-wise:} \quad \widehat{T}_{\mathrm{corr},j}(t) = \widehat{T}_{\mathrm{base},j}(t) + b_{\mathrm{local},j}(t), \tag{9.8}$$

$$\text{hybrid:} \quad \widehat{T}_{\mathrm{corr},j}(t) = \widehat{T}_{\mathrm{base},j}(t) + b_{\mathrm{common}}(t) + b_{\mathrm{local},j}(t). \tag{9.9}$$

The common correction represents a residual offset shared across several monitored nodes, such as broad environmental or model-alignment mismatch. The node-wise correction represents a channel-specific residual offset, such as a local calibration shift. The hybrid correction combines both terms: the common term handles coherent multichannel mismatch, while the local term handles bounded channel-specific mismatch. The local term is explicitly bounded because freely adapting node-specific correction has the greatest risk of learning away local fault evidence.

Correction updates are causal and gated. At each sample, the detector first computes the current prediction, residual, and normalized residual using only the correction state already available at that time. The correction state is then updated only for later samples if the update conditions are satisfied. Telemetry is treated as valid only when the temperature sample is available, finite, within the configured engineering range, and not masked by a gap or reset condition. Updates are also disabled near fast thermal-transition regions, where small phase or timing errors can produce structured residuals that should not be interpreted as a stable baseline offset.

The update gates are included to prevent adaptation during likely anomaly evidence. In particular, correction updates are restricted when $|z_{j,t}|$ is large, when telemetry is invalid, when the eligible residuals are not coherent across nodes, or when the local correction has reached its configured bound. This prevents the adaptive layer from immediately reducing the residual evidence that the downstream detector is intended to score.

For the robustness experiments in Chapter 12, the selected adaptive configuration is the hybrid bias-correction mode summarized in Table 9.4. The parameters were selected before final evaluation using validation-stage predictor-mismatch replays and local-fault retention checks. The selection objective was to reduce persistent mismatch-driven alert burden while retaining sensitivity to local thermal-fault evidence. The configuration is therefore carried forward as a proof-of-concept mismatch mitigation, not as a flight-ready adaptive correction policy.

Table 9.4: Selected hybrid adaptive correction parameters used in the predictor-mismatch robustness evaluation.

Parameter	Selected value	Role in the correction logic
Correction mode	Hybrid: $b_{\text{common}}(t) + b_{\text{local},j}(t)$	Combines a shared multichannel correction with a bounded node-specific correction.
Learning rate η_{sample}	0.007	Controls how quickly the correction moves toward eligible residual offsets. At a 15 s cadence, this gives slow correction over many samples rather than an immediate jump.
Residual update gate	$\lvert z_{j,t}\rvert \le 6.0$	Allows updates only when residual evidence is not already strongly anomaly-like. Larger residuals are scored by the detector but not immediately learned away.
Common-mode residual-spread limit	$\text{MAD}_{\text{common}} \le 0.75\,°\text{C}$	Allows common correction only when eligible node residuals are sufficiently coherent. Large spread suggests local behaviour rather than shared mismatch.
Maximum local correction	$\lvert b_{\text{local},j}(t)\rvert \le 0.5\,°\text{C}$	Limits how much node-specific adaptation can recentre an individual channel, reducing the risk of suppressing local fault evidence.

Figure 9.3 illustrates the intended role of the adaptive correction layer. Under a sustained predictor mismatch, the fixed predictor leaves a biased residual stream. The hybrid adaptive predictor gradually shifts the corrected prediction toward the observed baseline and reduces the post-settle residual mean. This is useful for persistent mismatch, but it also motivates slow, bounded, and gated updates so that large or local fault-like residuals are not immediately absorbed.

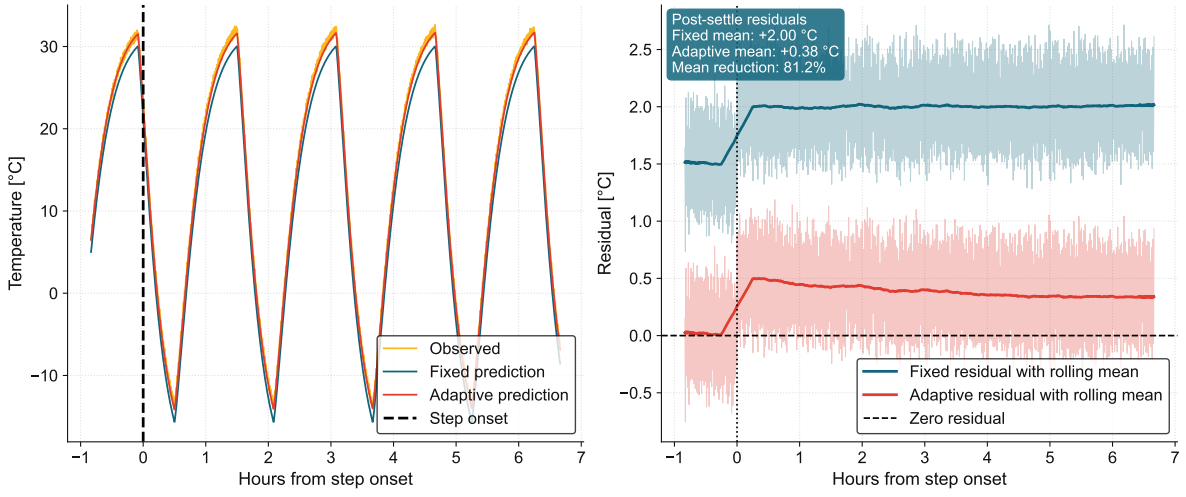


Figure 9.3: Example of adaptive residual-bias correction under sustained predictor mismatch.

9.7. Deployment Implications and Claim Boundaries

The fixed predictor is suitable as a compact expected-temperature reference for residual generation under matched nominal assumptions. A mission-ready on-board implementation, however, requires more than residual accuracy. The considerations in Table 9.5 should be addressed before the predictor can be considered flight-ready, with additional caution for the adaptive correction layer.

Table 9.5: On-board deployment considerations for the expected-temperature predictor.

Issue	Deployment implication
Runtime context availability	The predictor requires orbital phase, eclipse-transition context, and annual-term inputs to be available or reconstructed on board.
Phase and eclipse-transition sensitivity	Errors in θ_T or θ_E can create structured residuals, especially near eclipse ingress and egress.
Annual-term interpretation	The annual term is a geometric irradiance basis, not a measured solar-input estimate; solar-power telemetry could provide useful validation context.
Parameter storage	Template parameters, annual-term parameters, residual scalers, and detector thresholds require reliable storage, versioning, and corruption checks.
Residual normalization validity	The residual centre and scale define detector normalization; drift in nominal residual statistics can miscalibrate thresholds.
Mode dependence	A flight predictor may require mode-conditioned templates or context-dependent correction terms.
Adaptive correction risk	Adaptive correction terms must be slow, bounded, gated, and visible to operators to avoid absorbing fault evidence.
Embedded numerical implementation	The Microcontroller Unit (MCU) implementation should be compared against the Python reference because floating-point precision and approximations may differ.
Fallback behaviour	Missing, stale, or inconsistent context should trigger safe behaviour, such as skipping scoring, resetting detector memory, or issuing a telemetry-quality alert.

The considerations in Table 9.5 define what remains between the predictor demonstrated in this thesis and an operational flight implementation. Runtime context, including orbital phase, eclipse-transition state, and annual timing, must be available or reconstructed on board. Predictor parameters, residual normalization values, and detector thresholds must be stored reliably and checked for version consistency before use. Residual normalization must also remain valid as nominal behaviour changes over mission life. Before extending the detector beyond the `Nominal Mode` scope used here, mode-aware prediction or mode-conditioned correction would also be required.

The adaptive correction layer requires additional caution. It can reduce false alerts caused by predictor mismatch, but it can also reduce local residual evidence if it updates during anomaly onset or responds too quickly to abnormal behaviour. For this reason, the adaptive extension is evaluated as a robustness mechanism for predictor-mismatch experiments, not as an autonomous flight model-update strategy.

The predictor demonstrated here should therefore be interpreted as a compact expected-temperature reference for residual generation in the thesis prototype. It supports the event detector under the evaluated conditions, but it does not by itself establish long-term flight readiness.

9.8. Chapter Summary

This chapter selected and verified the expected-temperature predictor used by the residual detector. The predictor literature was reviewed in Subsection 2.8.2; this chapter applied that review to the Delfi Twin residual-to-alert pipeline. The fixed predictor carried forward is a two-branch stretched-exponential orbit-template model with annual modulation. It is intentionally compact and physically interpretable: it captures the dominant nominal thermal structure while leaving spin-imprinted ripple and nominal noise as residual variation for the detector to tolerate.

Residual-quality verification showed that the selected predictor removes the dominant orbit-scale thermal structure and produces centred residuals whose RMSE matches the expected floor from deliberately unmodelled spin-imprinted ripple and nominal noise. This supports its use as the fixed expected-temperature predictor under matched synthetic conditions.

The chapter also implemented a bounded adaptive correction layer for predictor-mismatch robustness. This extension can reduce residual bias under broad mismatch, but it is constrained by update gates and local bounds to reduce the risk of suppressing local anomaly evidence.

The predictor output from this chapter is passed to Chapter 10, where residuals are converted into instantaneous evidence, cumulative evidence, bounded events, and detector-faithful alert packets. Later evaluation chapters test how this predictor and decision layer perform under matched conditions, predictor mismatch, embedded replay, and real-telemetry stress cases.

10

Anomaly Decision and Explainability

Section 2.9 reviewed the score-to-event decision concepts and explainability requirements relevant to deployable spacecraft telemetry anomaly detection. Chapter 8 then selected a residual-to-alert architecture for the Delfi Twin temperature-telemetry use case, and Chapter 9 defined the expected-temperature predictor used to generate residual evidence. This chapter implements the selected decision layer.

The purpose of the decision layer is to convert normalized temperature residuals into bounded, operator-readable alert events. The layer performs 4 functions. First, it converts signed residuals into instantaneous and cumulative evidence. Second, it combines this evidence into sample-level candidate alarms. Third, it stabilizes candidate alarms through event-lifecycle logic, including persistence, hysteresis, quiet reset, transient-spike suppression, and gap termination. Fourth, it packages the resulting event into a compact detector-faithful alert payload, defined later in Section 10.6.

The contribution is their causal composition into a residual-to-alert lifecycle suitable for Microcontroller Unit (MCU)-class implementation and compact downlink. The lifecycle handles persistence, recovery, gaps, and transient evidence while preserving detector-faithful evidence for ground review. It does not perform root-cause diagnosis on board; diagnosis, response selection, and model updates remain ground responsibilities.

Deployment-Oriented Workflow Steps

5. Screen and down-select the end-to-end pipeline
Evaluate common approaches based on the characteristics of the telemetry and specific application, dismissing incompatible techniques.

6. Tune and evaluate the selected pipeline
Benchmark candidate method families as end-to-end pipelines, including preprocessing and score-to-alert logic, using the protocol defined above.

10.1. Selected Residual-to-Alert Pipeline

Figure 10.1 summarizes the decision layer implemented in this chapter. The input is the normalized signed residual produced by comparing observed temperature telemetry with the expected-temperature predictor from Chapter 9. The output is a bounded detector event and compact detector-faithful alert payload, whose fields are defined in Section 10.6.

The decision layer is organized around three implementation requirements. First, it must provide fast evidence for large abrupt deviations, such as calibration steps, telemetry-quality anomalies, and rapid thermal excursions. Second, it must accumulate evidence for smaller persistent deviations, such as heater faults, self-heating anomalies, and slow residual shifts. Third, it must stabilize the alert stream so that isolated spikes, near-threshold noise, stale cumulative evidence, and telemetry gaps do not produce misleading events.

The following sections define this pathway in implementation order: residual normalization, instantaneous evidence, cumulative directional evidence, candidate-alarm formation, event-lifecycle logic, and detector-faithful alert packaging.

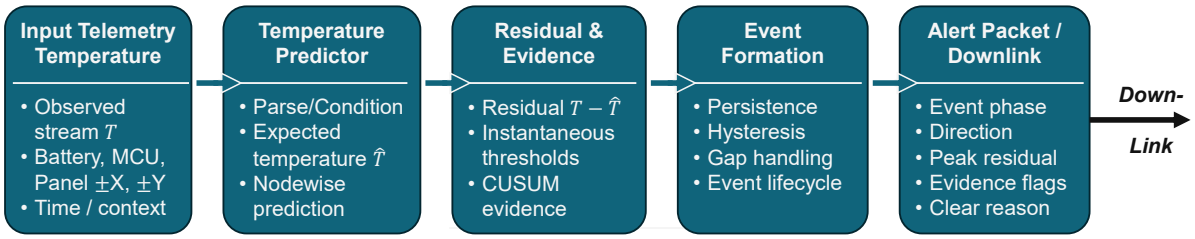


Figure 10.1: Selected residual-to-alert pipeline from expected-temperature prediction to residual evidence, event lifecycle logic, and compact detector-faithful alert packets.

10.2. Residual Normalization and Signed Evidence

The raw residual definition and sign convention were introduced in Section 9.1. In this chapter, the residual is converted into the normalized evidence signal used by the decision layer. For each temperature channel j, the raw residual is standardized using nominal residual statistics estimated from fault-free training data:

$$z_{j,t} = \frac{r_{j,t} - \mu_{r,j}}{\sigma_{r,j}}. \tag{10.1}$$

Here, $\mu_{r,j}$ and $\sigma_{r,j}$ denote the channel-wise nominal residual mean and standard deviation. This normalization allows a shared detector configuration to be applied across battery, MCU, and panel temperature nodes, despite differences in residual scale.

The residual sign is preserved after normalization. Positive values indicate hotter-than-expected behaviour, while negative values indicate cooler-than-expected behaviour. This signed normalized residual is the input to both the instantaneous threshold logic and the positive and negative cumulative evidence streams defined in the following sections.

10.3. Instantaneous and Cumulative Evidence

Chapter 8 selected parallel instantaneous and cumulative evidence as part of the residual-to-alert architecture. This section implements those two evidence streams for the signed normalized residuals defined in Section 10.2. The output of this section is not yet a final alert event; it is sample-level evidence that is passed to the event-lifecycle logic in Section 10.5.

Instantaneous Evidence

Instantaneous evidence provides a fast response to large abrupt residual excursions. A positive instantaneous alarm is raised when the normalized residual exceeds the positive threshold, and a negative instantaneous alarm is raised when it falls below the corresponding negative threshold:

$$z_{j,t} > \tau_z, \qquad z_{j,t} < -\tau_z. \tag{10.2}$$

This evidence stream is most useful when the deviation is already large at the sample level, such as an abrupt calibration step, severe telemetry-quality anomaly, or rapid thermal excursion. It is less suitable on its own for gradual thermal faults, where each individual residual may remain below the instantaneous threshold even though the deviation becomes meaningful over time.

Cumulative Evidence

Cumulative evidence tracks sustained directional residual shifts. The detector maintains one evidence stream for hotter-than-expected behaviour, $C^{+}_{j,t}$, and one for cooler-than-expected behaviour, $C^{-}_{j,t}$:

$$C^{+}_{j,t} = \max\left(0,\ C^{+}_{j,t-1} + z_{j,t} - k_{\text{CUSUM}}\right), \tag{10.3}$$

$$C^{-}_{j,t} = \max\left(0,\ C^{-}_{j,t-1} - z_{j,t} - k_{\text{CUSUM}}\right). \tag{10.4}$$

The slack parameter k_{CUSUM} acts as a per-sample allowance for nominal residual variation. Only residual evidence beyond this allowance contributes to the cumulative sum. This prevents small, ordinary residual fluctuations from accumulating indefinitely, while still allowing persistent directional deviations to build evidence over time. A cumulative alarm is raised when either directional evidence stream exceeds its decision limit:

$$C^{+}_{j,t} > h_{+} \qquad \text{or} \qquad C^{-}_{j,t} > h_{-}. \tag{10.5}$$

The two-sided form preserves the physical meaning of the residual sign. Excess self-heating and battery over-temperature contribute to positive evidence, while heater-off behaviour or missing expected self-heating contributes to negative evidence. These instantaneous and cumulative alarms are combined in the next stage to form candidate alarms before event-lifecycle rules determine whether a stable bounded event should be declared.

10.4. Candidate-Alarm Logic

The detector first converts the residual evidence for each channel into a sample-level candidate-alarm flag. A sample is considered alarming if either the instantaneous normalized residual exceeds the configured threshold or the positive or negative Cumulative Sum (CUSUM) evidence exceeds its decision limit:

$$a_{j,t} = \left(z_{j,t} > \tau_z\right) \vee \left(z_{j,t} < -\tau_z\right) \vee \left(C^{+}_{j,t} > h_{+}\right) \vee \left(C^{-}_{j,t} > h_{-}\right). \tag{10.6}$$

This combined rule preserves rapid response to large deviations while allowing smaller persistent residuals to accumulate into reportable evidence. It also supports detector-faithful explainability because the detector can record which evidence source contributed to event onset and which evidence sources appeared during the event.

The candidate-alarm flag is the first combined alarm signal in the residual decision layer. It reduces the instantaneous and cumulative evidence streams to one sample-level alarm stream, while preserving the underlying evidence flags for later alert reporting.

10.5. Event Lifecycle Composition

Candidate alarms are not downlinked directly as alert events. Instead, they are passed through an event lifecycle that converts sample-level evidence into stable bounded events. The lifecycle uses established alert-stabilization primitives, including persistence, hysteresis, reset logic, and gap handling. This chapter defines how those primitives are connected and applied in the Delfi Twin residual detector.

Figure 10.2 shows the event lifecycle. The detector begins in an inactive monitoring state. Candidate evidence moves the detector into an alarm-building state, but an event is opened only after the start-persistence rule is satisfied. Once active, the event remains open until the clear-hysteresis rule is satisfied, a telemetry gap terminates observability, or the replay window ends before either condition occurs. Table 10.1 summarizes the lifecycle rules and their operational purpose.

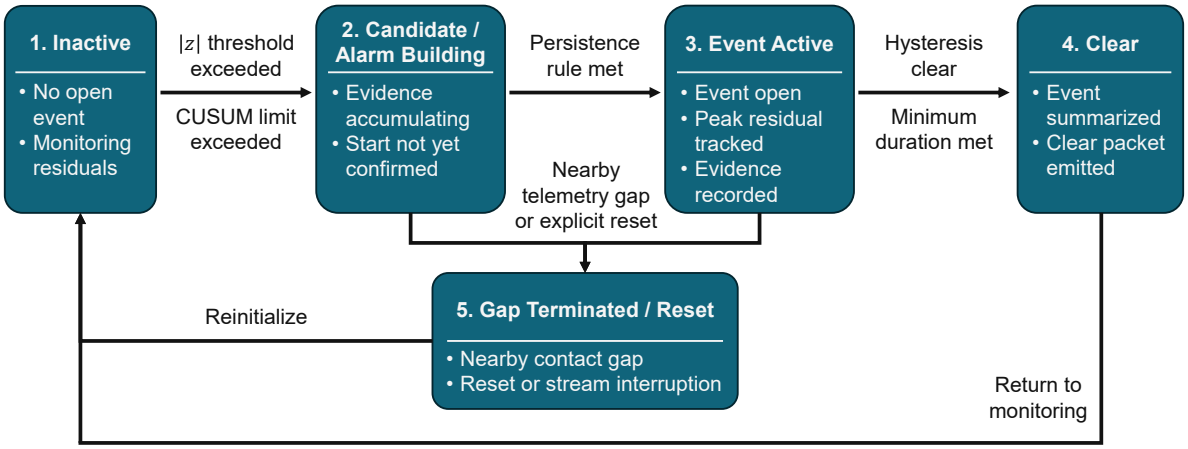


Figure 10.2: Residual decision-layer event lifecycle, showing event start, hysteresis clearing, gap termination, and open-window handling.

Table 10.1: Event lifecycle rules and their operational purpose.

Rule / parameter	Lifecycle role	Purpose
Candidate alarm $a_{j,t}$	Creates a sample-level candidate-alarm flag.	Combines instantaneous threshold and cumulative CUSUM evidence into one alarm stream.
Threshold τ_z	Contributes to candidate-alarm formation.	Provides rapid response to large single-sample residual deviations.
CUSUM limits h^+, h^-	Contribute to candidate-alarm formation.	Allow smaller sustained positive or negative residuals to accumulate into reportable evidence.
Start rule k_s of n_s	Promotes *Candidate / Alarm Accumulating* to *Event Active*.	Requires persistence before opening an event, reducing sensitivity to isolated alarm samples.
Clear rule k_c of n_c	Moves *Event Active* to *Event Closed*.	Provides hysteresis by requiring sustained quiet behaviour before clearing.
Minimum event duration	Applied before retaining an event for reporting.	Prevents very short fragments from becoming downlinked alert events.
Quiet threshold τ_{quiet}	Identifies near-nominal residual behaviour.	Allows stale CUSUM evidence to be reset after the residual returns to the quiet band.
Transient-spike suppression	Acts during the early candidate stage.	Rejects short point-like disturbances that return quickly to the quiet band.
Telemetry gap rule	Moves candidate or active states to *Gap Terminated / Reset*.	Terminates evidence when observability is lost, without claiming normal recovery.
`open_at_end`	Offline replay label for unfinished active events.	Records that the data window ended before normal clearing or gap termination occurred.

The lifecycle separates three levels of decision-making. First, threshold and CUSUM rules produce sample-level candidate evidence. Second, persistence and hysteresis decide when events open and close. Third, quiet reset, transient-spike suppression, and telemetry-gap handling prevent residual evidence from carrying forward when the signal has returned to near nominal, when the disturbance is only transient, or when telemetry is no longer observable.

10.5.1. Event Start Persistence

An event is declared active only when the sample-level alarm evidence is persistent. Specifically, at least (k_s) of the last (n_s) samples must satisfy the alarm condition:

$$\sum_{\ell=t-n_s+1}^{t} \mathbb{1}\left(a_{j,\ell}\right) \geq k_s. \tag{10.7}$$

This suppresses isolated single-sample threshold crossings and ensures that an event is opened only when residual evidence persists. In the lifecycle diagram, this rule governs the transition from *Candidate / Alarm Accumulating* to *Event Active*.

10.5.2. Event Clear Hysteresis

Once an event is active, it is not cleared immediately at the first quiet sample. Instead, the clear rule requires at least k_c of the last n_c samples to be non-alarming:

$$\sum_{\ell=t-n_c+1}^{t} \mathbb{1}\left(\neg a_{j,\ell}\right) \geq k_c. \tag{10.8}$$

This hysteresis prevents alert chatter when residuals hover near the decision threshold. It also prevents a single quiet sample from closing an event that may still be active. The event is therefore cleared only after sustained evidence that the residual has returned near nominal. In the lifecycle diagram, this rule governs the transition from *Event Active* to *Event Closed*.

10.5.3. Minimum Event Duration and Open Events

Short event fragments are suppressed using a minimum-duration rule. An event must remain active for at least the configured minimum number of samples before it is retained for reporting. This prevents brief fragments from becoming operator-facing events.

During offline replay and evaluation, an event may remain active when the available data window ends. In this case, the event is assigned the clearance reason `open_at_end`. This does not imply physical recovery or loss of observability. It only indicates that the replay window ended before the lifecycle reached either a hysteresis clear or a gap-terminated state.

10.5.4. Quiet-Evidence Reset

A practical issue with cumulative evidence is that it can remain elevated after a transient event, even when the residual has returned close to nominal. This behaviour can be seen in Figure 10.3a. If left untreated, stale CUSUM evidence can keep an event active or make the detector overly sensitive long after the original deviation has ended.

To prevent this, the detector includes a quiet reset. If the absolute normalized residual remains close to zero for a configured number of consecutive valid samples, the cumulative evidence terms are reset:

$$\left|z_{j,t}\right| < \tau_{\text{quiet}} \quad \Rightarrow \quad C^{+} * j,t \leftarrow 0, C^{-} * j,t \leftarrow 0. \tag{10.9}$$

Here, τ_{quiet} defines how close to zero the normalized residual must remain before the detector treats the signal as quiet. This reset does not directly force an event to clear. Instead, it removes stale cumulative evidence so that the clear-hysteresis rule reflects the current residual state. Quiet reset is therefore a supporting lifecycle rule, not a separate declaration that the event has recovered. This functionality is demonstrated in Figure 10.3b.

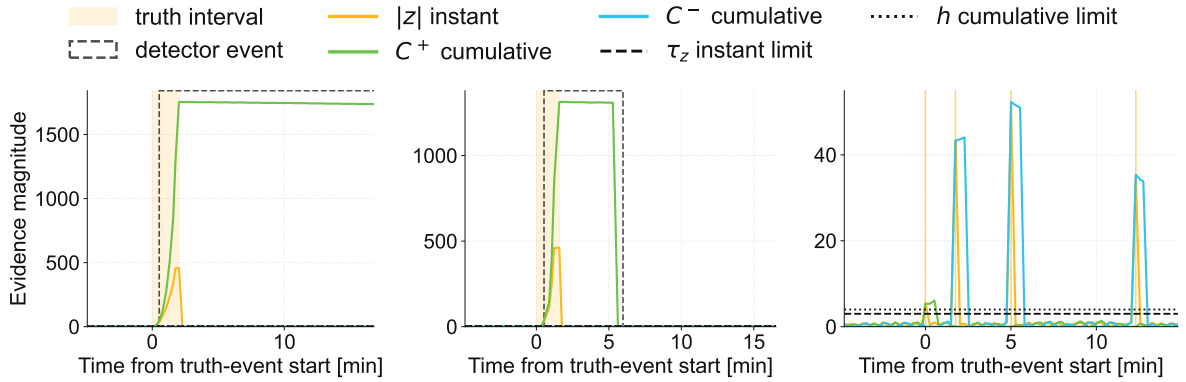
10.5.5. Transient-Spike Suppression

Temperature telemetry can contain isolated point-like disturbances that should not be interpreted as alert-worthy events. Start-persistence and minimum-duration rules reduce sensitivity to these disturbances, but they act after candidate evidence has already been generated. A large single-sample residual can still inject cumulative evidence into the detector state, even if the residual quickly returns close to nominal. The detector therefore includes a transient-spike suppressor, demonstrated in Figure 10.3c.

A short initial threshold excursion is treated as suppressible if it lasts no longer than a configured maximum initial duration and is followed by a short confirmation window in which the residual returns close to zero. In the embedded implementation, this is handled causally: the detector temporarily holds the candidate excursion while waiting for quiet confirmation. If the residual returns to the quiet range within the confirmation window, the candidate alarm is suppressed and cumulative evidence is reset. If the residual remains active, the event proceeds through the normal persistence and cumulative-evidence logic.

This mechanism reflects the benchmark definition of isolated spikes as benign telemetry nuisances rather than thermal-fault evidence. The minimum-duration rule limits which event fragments are retained for reporting, while transient-spike suppression prevents isolated point disturbances from carrying forward into cumulative evidence and event-lifecycle state.

Figure 10.3 brings these examples together for comparison. The panels are thesis-generated decision-behaviour illustrations, not external benchmark results.



(a) Stale cumulative evidence after a transient when no quiet reset is applied.

(b) Quiet reset after sustained near-nominal residuals.

(c) Transient spike suppression for a single-sample impulse followed by quiet confirmation.

Figure 10.3: Decision-evidence examples illustrating quiet reset and transient-spike suppression.

10.5.6. Telemetry Gaps and Gap Termination

Telemetry gaps represent loss of observability. If the detector is inactive, a gap resets cumulative evidence and persistence state. If a gap occurs while an event is active, the event is closed with the clearance reason `gap_terminated` rather than with a normal hysteresis clear.

This policy is operationally important. A gap does not prove that the anomaly continued, and it also does not prove that the anomaly recovered. Therefore, the event is terminated because observability was lost, and recovery is not claimed. After gap termination, the detector returns to the inactive state and begins a new evidence sequence when valid telemetry resumes.

10.6. Detector-Faithful Explainability and Alert Payload

Subsection 2.9.4 reviewed explainability as a deployment requirement for spacecraft telemetry Anomaly Detection (AD). In this thesis, explainability is implemented in detector-faithful form. The alert reports only quantities already produced by the residual decision layer: residual magnitude, event direction, evidence flags, event timing, and clearance reason. No separate post-hoc explanation model is added.

The explainability layer is designed to answer 4 ground-operator questions:

1. **Where did the event occur?**
   The affected node or channel is reported.
2. **What did the residual do?**
   The event direction, peak residual, peak normalized residual, and end residual are reported.

3. **Why was the event declared?**
   Evidence flags identify which detector mechanisms contributed to event declaration.
4. **How did the event end?**
   The clearance reason indicates whether the event cleared normally, was gap-terminated, or remained open at the end of the available window.

The on-board packet stores this information as compact numeric fields and encoded flags. A richer operator-readable message can then be generated on the ground by decoding those fields. This preserves embedded feasibility while maintaining a traceable path from detector state to operator display.

The payload is organized into 3 groups: lifecycle fields, residual summaries, and detector evidence. These fields are summarized in Table 10.2. Chapter 11 then describes how the same information is encoded in the embedded event packet.

Table 10.2: Compact event payload fields used for detector-faithful explainability.

Field group	Fields	Purpose
Lifecycle	`phase`, `event_id`, `node`, `start_sample`, `length_samples`, `clear_reason`	Identifies where the event occurred, when it started, how long it lasted, and why it ended.
Residual summary	`peak_abs_z`, `peak_signed_z`, `end_signed_z`, `peak_residual_C`, `end_residual_C`, `event_direction`, `peak_direction`	Summarizes the magnitude, direction, and final state of the temperature deviation.
Detector evidence	`n_alarm_samples`, `peak_cpos`, `peak_cneg`, `detector_flags_onset`, `detector_flags_any`	Explains which detector mechanisms contributed to declaration and what evidence appeared during the full event.

10.6.1. Onset Evidence and Audit Evidence

The alert payload reports two evidence summaries: onset evidence and audit evidence. Onset evidence records which detector evidence sources were active when the event was declared. Audit evidence records which evidence sources appeared at any point during the event.

These two fields serve different review purposes. An event may be declared after cumulative evidence crosses its threshold, while an instantaneous threshold crossing or telemetry gap occurs later in the same event. Onset evidence preserves the original declaration reason, while audit evidence captures the broader detector context observed during the event. This helps the ground operator interpret both why the event opened and what additional evidence appeared before it cleared or was terminated. The corresponding evidence flags are summarized in Table 10.3.

Table 10.3: Detector evidence flags used in offline and embedded event summaries.

Flag	Meaning
`DETECT_THRESH_POS`	Positive instantaneous threshold evidence.
`DETECT_THRESH_NEG`	Negative instantaneous threshold evidence.
`DETECT_CUSUM_POS`	Positive cumulative evidence exceeded its decision limit.
`DETECT_CUSUM_NEG`	Negative cumulative evidence exceeded its decision limit.
`DETECT_PERSISTENCE_START`	Start persistence rule was satisfied.
`DETECT_HYSTERESIS_CLEAR`	Event cleared by the hysteresis clear rule.
`DETECT_MIN_DURATION`	Event satisfied the minimum-duration requirement.
`DETECT_SPIKE_SUPPRESSED`	Transient spike suppression affected the event or evidence stream.
`DETECT_GAP_NEARBY`	Gap or reset evidence occurred near or during the event.
`DETECT_GAP_AT_ONSET`	Event onset was limited by nearby gap or reset evidence.

10.6.2. Event Direction, Peak Direction, and Clearance Reason

The event payload includes both event direction and peak direction. Event direction summarizes the dominant residual sign over the event, while peak direction records the sign of the largest absolute residual. For mixed-sign events, a temperature stream may first become hotter than expected and later become cooler than expected. Reporting both fields preserves the overall event direction as well as the direction of the strongest residual evidence.

The clearance reason records how the detector event ended. The implemented clearance reasons are:

- `hysteresis_clear`: residual evidence returned to a quiet state and the clear rule was satisfied;
- `gap_terminated`: telemetry observability was lost while the event was active;
- `open_at_end`: the available evaluation or replay window ended before the event cleared.

Contact gaps require explicit handling because they interrupt observability. If telemetry is lost during an active event, the detector cannot determine whether the anomaly recovered or persisted during the unobserved interval. The event is therefore closed as `gap_terminated`. This prevents artificially long event durations across missing telemetry while preserving the information that normal recovery was not directly observed.

10.6.3. Scope of Explainability

The explainability provided by this payload is scoped to detector evidence. It is detector-faithful, not diagnostic. It explains what the residual decision layer observed, not the physical root cause of the spacecraft behaviour. For example, the alert can report that the battery node was hotter than expected, that the event was driven by positive cumulative evidence, that the peak residual magnitude reached a specified value, and that the event cleared by hysteresis or was gap-terminated. It cannot, by itself, prove whether the underlying cause was a heater fault, battery fault, sensor calibration issue, or external environmental change.

This is consistent with the responsibility allocation defined in Section 3.2. The on-board system detects suspected events, summarizes detector evidence, and packages compact alert information. Ground operators retain responsibility for diagnosis, response selection, and configuration management. The alert payload is therefore designed to support ground-operator prioritization and review, not to replace operator judgment.

10.7. Validation Tuning and Sensitivity Selection

The detector configuration was selected using the validation split before final evaluation and embedded replay. Here, detector configuration refers to the thresholds and lifecycle parameters that control residual-to-event conversion: the instantaneous threshold, CUSUM slack and limits, start persistence, clear hysteresis, minimum event duration, quiet reset, and transient-spike suppression.

The tuning objective prioritized recovery of alert-worthy telemetry-quality anomalies and thermal faults, while penalizing excessive Time to Awareness (TTA), false alerts on benign nuisances, and event fragmentation. This reflects the intended operational role of the detector. Missing alert-worthy events would reduce the value of early anomaly awareness, but excessive sensitivity would create too many low-value alerts and reduce the usefulness of the limited downlink allocation. The selected configuration therefore balances event recovery, TTA, nuisance rejection, and compact alert value.

The validation sweep evaluated candidate detector configurations on the validation split. For each configuration, the main quantities were positive-event recall, P95 TTA, false alerts per day, nuisance-trigger behaviour, and event fragmentation. The final configuration was selected from the high-recall region by balancing time to awareness against alert burden.

Figure 10.4 summarizes the main validation trade-off between false alerts per day and P95 TTA. Each point represents one detector configuration. The preferred region contains configurations with full positive-event recall, low P95 TTA, and low false-alert burden. Because the sweep varied multiple detector parameters and compared each configuration using the same validation objective, it also provides a parameter-sensitivity analysis. The top-ranked configurations form a small cluster rather than a single isolated point, indicating that the selected setting is not a fragile single-configuration optimum.

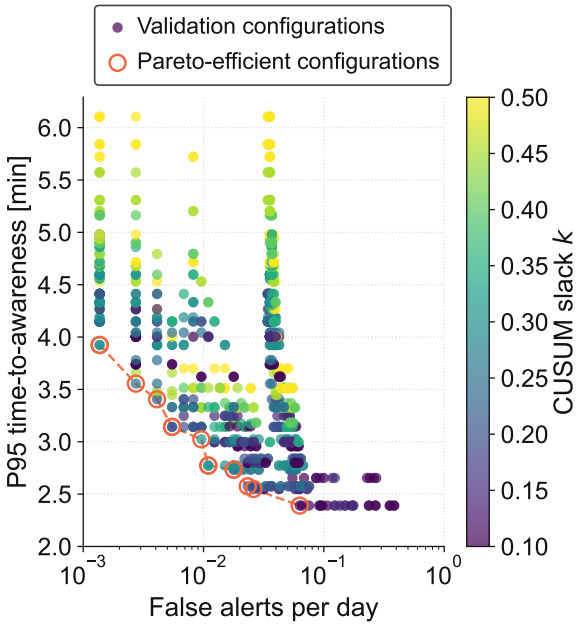


Figure 10.4: Validation trade-off between false-alert burden and p_{95} time-to-awareness across detector configurations.

Additional parameter-sensitivity plots and threshold heatmaps are provided in the tuning appendix. These show how the CUSUM decision limit affects false-alert burden and TTA, and how persistence settings trade off sensitivity against nuisance rejection.

The final configuration was therefore selected from a cluster of near-equivalent top validation configurations, favouring a simple and interpretable setting that transfers cleanly to the embedded implementation. The selected high-performing configurations achieved full positive-event recall, P95 TTA of about 1.82 min, and false-alert burden around 0.074 to 0.077 false alerts per day.

10.8. Selected Decision Configuration

The selected detector configuration was fixed after validation and carried forward to the evaluation split and embedded replay. Here, detector configuration refers to the thresholds and lifecycle parameters that control residual-to-event conversion. The final values are summarized in Table 10.4.

Together, these parameters reflect the validation trade-off between event recovery, TTA, nuisance rejection, and event fragmentation. The instantaneous threshold provides rapid response to high-magnitude deviations, while CUSUM captures sustained lower-amplitude residual shifts. Persistence, hysteresis, minimum duration, quiet reset, transient-spike suppression, and gap termination then convert this evidence into stable bounded alert events.

The selected configuration should be interpreted as the matched-predictor decision configuration. It is used under the fixed-predictor assumptions established in Chapter 9 and evaluated under matched nominal conditions before being tested under predictor mismatch. It does not by itself establish mission-long robustness if the residual baseline drifts. Later robustness evaluation therefore tests how this same configuration behaves when the observed telemetry no longer matches the fixed nominal predictor.

10.9. Chapter Summary

This chapter defined the residual decision and explainability layer used in the thesis. The selected detector combines instantaneous threshold evidence, signed cumulative evidence, persistence, hysteresis, minimum event duration, quiet reset, transient-spike suppression, and telemetry-gap termination. These mechanisms convert normalized residual streams into stable bounded alert events.

Table 10.4: Selected residual decision configuration used for offline evaluation and embedded implementation.

Parameter	Value	Role
(τ_z)	3.0	Instantaneous normalized-residual threshold for large single-sample deviations.
CUSUM (k_{CUSUM})	0.25	Per-sample allowance before residual evidence contributes to cumulative growth.
(h_+), (h_-)	4.0, 4.0	Positive and negative cumulative-evidence decision limits.
Start rule ((k_s,n_s))	((2, 3))	Opens an event when 2 of the last 3 samples satisfy the candidate-alarm condition.
Clear rule ((k_c,n_c))	((3, 3))	Clears an active event after 3 quiet samples in a 3-sample window.
Minimum event duration	5 samples	Excludes very short event fragments from reporting.
Quiet reset threshold/window	1.0, 8 samples	Resets cumulative evidence after ($\lvert z_{j,t}\rvert$<1.0) for 8 consecutive valid samples.
Gap-termination samples	1 sample	Terminates an active event when telemetry observability is lost.
Spike quiet threshold/window	1.5, 3 samples	Requires return close to zero for 3 samples before suppressing an initial spike.
Maximum initial spike length	1 sample	Restricts transient-spike suppression to one-sample initial excursions.

The alert payload is detector-faithful. It reports residual summaries, event direction, peak direction, clearance reason, onset evidence, and event-wide audit evidence. It does not perform root-cause diagnosis on board. Instead, it provides compact detector-faithful evidence that can be decoded on the ground into alerts for ground-operator review.

The final detector configuration was selected using the validation split and fixed before evaluation and embedded replay. Chapter 11 implements this decision layer on the target embedded platform, while Chapter 12 evaluates the selected configuration on the previously unused synthetic evaluation split and under predictor-mismatch stress cases.

Part III:

Implementation, Verification, and Experimental Results

Building on the benchmark and method design established in Part II, this part implements the residual-to-alert pipeline and evaluates its behaviour experimentally. It presents the STM32L4-class prototype, offline and embedded verification evidence, synthetic benchmark results, predictor-mismatch analysis, and real-telemetry stress tests needed to assess prototype performance and embedded feasibility.

11

Embedded Prototype Implementation

This chapter describes the STM32L4-class prototype implementation of the selected thermal Anomaly Detection (AD) pipeline. The goal is not to implement the complete Delfi Twin flight-software stack. Instead, the objective is to determine whether the selected expected-temperature predictor, residual evidence, event-lifecycle logic, adaptive correction path, and compact event-packet pathway can execute causally on representative low-power microcontroller hardware.

The focus is on implementation boundaries, embedded state representation, replay interfaces, host-to-target verification, agreement with the offline Python reference, and timing, memory, and packet-size instrumentation. Numerical feasibility results are reported in Chapter 12. This separation keeps the present chapter focused on how the embedded prototype was constructed and verified, while the evaluation chapter assesses whether the implementation satisfies the prototype requirements.

Deployment-Oriented Workflow Steps

7. Implement and verify under embedded constraints
Implement the selected pipeline on the target platform, demonstrate causal operation (no future samples), profile timing/memory, and validate robustness on the defined scenario suite.

11.1. Implementation Objective and Boundary

The embedded implementation translates the selected residual-to-alert pipeline into a C firmware prototype with bounded state and fixed packet outputs. The implemented core includes on-board thermal prediction, residual normalization, instantaneous and cumulative evidence, event-lifecycle management, transient-spike suppression, telemetry-gap termination, node-wise adaptive bias correction, event-packet generation, and diagnostic timing output.

The prototype is narrower than a flight-software implementation. It does not implement the full spacecraft telemetry task, on-board storage service, downlink scheduler, configuration-management workflow, or spacecraft mode-management layer. Instead, replay samples are supplied by a host computer over UART. The host selects event windows or validation streams, formats one input packet per sample, receives embedded diagnostic and event packets, and compares the board output with the offline reference implementation. The UART interface is therefore a verification interface, not the intended flight telemetry interface.

Table 11.1 summarizes the boundary between the embedded detector core, the host-side verification harness, and the future flight-software integration layer.

This boundary is important for interpreting the embedded results. The prototype supports claims about causal execution, memory use, timing margin, packet size, and offline-to-embedded agreement under replay. It does not by itself prove integrated flight-software schedulability, flight-board power consumption, radiation robustness, operational configuration safety, or end-to-end downlink handling.

Table 11.1: Embedded prototype boundary between the STM32 firmware, host-side replay tooling, and future flight-software integration.

Element	Role in this prototype	Flight interpretation
Detector core	Runs on-board prediction, residual scoring, evidence update, lifecycle logic, and event-packet generation on the STM32.	Reusable embedded AD component after integration with flight telemetry, storage, and downlink services.
UART replay interface	Streams controlled samples and context fields from the host to the board.	Verification interface only; would be replaced by the spacecraft telemetry interface.
Host-side replay scripts	Select event windows or validation streams, build board input rows, and send one sample packet at a time.	Ground verification infrastructure, not flight software.
Event packets	Encode compact detector-faithful event packets emitted by the firmware.	Prototype representation of the intended on-board alert product.
Diagnostic and verbose packets	Expose timing, interval state, and per-sample debug fields during replay.	Useful for testing and commissioning; not necessarily downlinked during nominal operations.

11.2. Target Platform and Build Context

The target platform is an STM32L4-class Microcontroller Unit (MCU) prototype. This class reflects the low-power MCU environment relevant to the Delfi Twin On-Board Computer (OBC) context introduced in Section 3.4. The firmware is implemented in the C programming language and uses static configuration and detector-state structures. Dynamic memory allocation is avoided in the detector path so that memory use remains bounded and measurable.

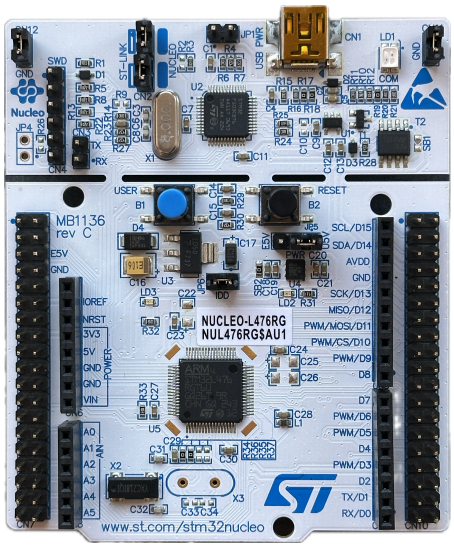


Figure 11.1: STM32L476RG development board used for the embedded prototype.

Timing measurements are based on firmware cycle-count instrumentation reported through diagnostic packets. Cycle counts are converted to time using the reported board clock. The resulting values characterize replay execution of the detector core on the prototype board. They should not be interpreted as formal Worst-Case Execution Time (WCET) proof for an integrated flight-software task.

Memory use is measured from the compiled firmware build artifacts. The primary memory result is the whole-firmware flash and RAM footprint. Linker-map inspection is used only as supporting evidence to understand approximate object-level contributions, because shared library code, compiler optimization, and call-site effects can make exact attribution to individual modules unreliable.

Direct flight-board power was not measured. The evaluation therefore reports a compute-duty estimate based on measured processing time relative to the 15 s telemetry cadence. This is sufficient to assess whether compute demand is plausibly negligible at prototype level, but it is not a substitute for current or energy measurement on the integrated flight board.

11.3. Embedded Software Architecture

The firmware implements the residual-to-alert pipeline developed in the preceding chapters, but the emphasis here is implementation rather than detector formulation. The expected-temperature predictor, residual evidence, cumulative evidence, persistence, hysteresis, transient-spike suppression, and gap-handling rules were defined earlier. This chapter focuses on translating those definitions into static, causal, per-node firmware state with bounded packet outputs, and on constructing a replay harness that allows the firmware implementation to be compared with the offline Python reference.

Figure 11.2 summarizes the resulting host-to-STM32 replay and verification workflow. The left-hand side shows how host-side scripts select event windows, validation streams, or predictor-mismatch scenarios and convert them into one board row per sample. These rows are formatted as UART input packets and streamed to the board. The centre block shows the STM32 firmware core, which parses each replay packet, performs prediction and residual-to-event processing, and emits event, diagnostic, and verbose packets. The right-hand side shows the host-side analysis path, where returned packets are decoded and used for offline-to-embedded agreement analysis and replay-performance summaries.

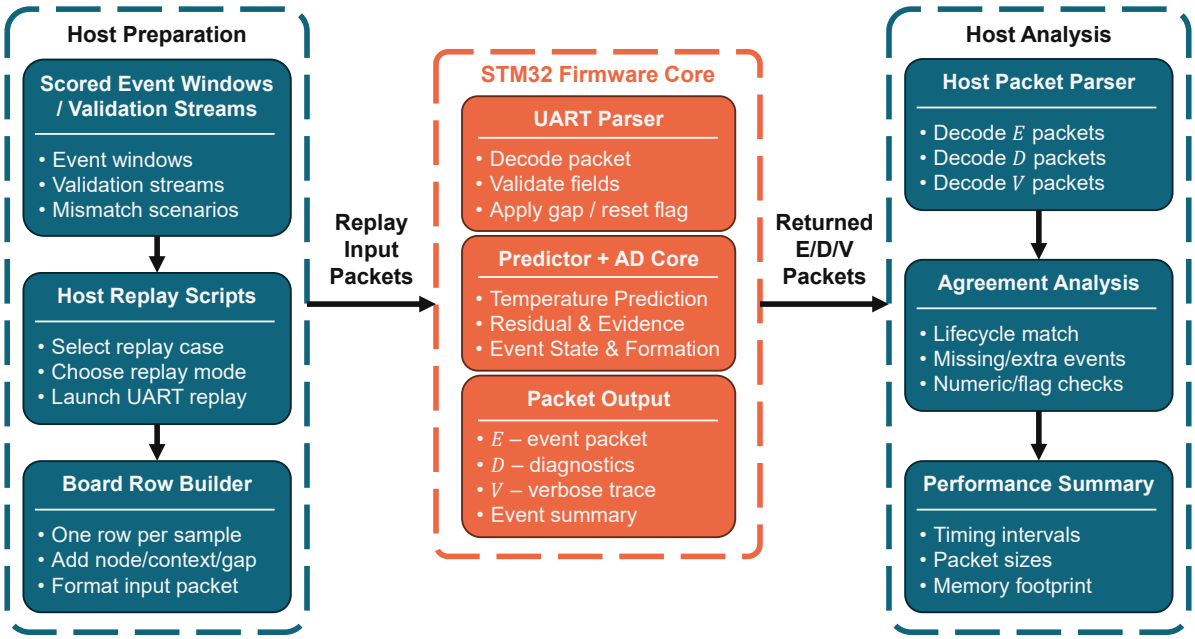


Figure 11.2: Host-to-STM32 replay and embedded verification workflow used for causal detector replay, packet decoding, agreement analysis, and resource measurement.

The firmware core in Figure 11.2 is organized into the modules summarized in Table 11.2. The table clarifies the implementation role of each component without repeating the decision-logic derivation.

This modular structure supports two forms of evidence. First, it demonstrates that the selected decision layer can be implemented as bounded embedded software. Second, it provides the observability needed to compare embedded behaviour with the offline reference implementation and to measure timing, memory, and packet-size behaviour under controlled replay.

11.4. Static State and Causal Execution

The firmware processes telemetry one sample at a time. It does not access future samples, batch windows, or offline labels during detector execution. This is essential because the on-board detector must make lifecycle decisions using only the state available at the current sample.

Table 11.2: Embedded firmware modules and their implementation roles.

Firmware module	Embedded responsibility	Reason for inclusion
Input parser	Decodes replay packets containing sample index, node identity, observed temperature, orbital context, and gap flags.	Replaces the spacecraft telemetry interface during prototype testing while preserving causal sample-by-sample execution.
Prediction module	Computes fixed on-board expected temperature or node-wise adaptive corrected prediction from stored parameters.	Tests whether prediction can be performed on the target MCU rather than supplied by Python.
Residual and evidence module	Computes signed residuals, normalized residuals, instantaneous threshold evidence, cumulative evidence, and evidence flags.	Preserves the selected detector semantics in causal embedded execution.
Event-state module	Maintains persistence, hysteresis, transient-spike suppression, gap termination, active-event state, and event summaries.	Converts pointwise residual evidence into bounded lifecycle events without future samples.
Packet formatter	Emits event, diagnostic, and verbose packets.	Provides compact alert evidence, timing diagnostics, and replay-debug information.
Timing instrumentation	Records cycle counts over diagnostic intervals.	Supports timing and duty-cycle evaluation in Chapter 12.

The embedded detector stores one state structure per monitored temperature node. This state contains the current residual and normalized residual, positive and negative cumulative evidence, active-event status, event start sample, event duration, alarm-sample count, peak residual summaries, event direction, peak direction, evidence flags, clear counter, clearance reason, gap context, and transient-spike confirmation state. When adaptive prediction is enabled, the node-wise adaptive bias state is also stored and updated causally.

This state does not reproduce the full offline analysis environment. Instead, it stores the information needed for the embedded detector to start, continue, clear, suppress, or gap-terminate an event, while also supplying the fields needed for compact event packets. This state representation is where the decision logic from Chapter 10 becomes bounded firmware behaviour for the embedded feasibility evaluation in Chapter 12.

Table 11.3 summarizes the main embedded state classes and their role.

Table 11.3: Main per-node state classes used by the embedded detector.

State class	Examples	Purpose
Prediction state	Fixed model parameters, residual scaler, node-wise adaptive bias.	Computes the expected temperature and normalized residual evidence.
Evidence state	Current residual, normalized residual, C^+, C^-, instantaneous and cumulative evidence flags.	Maintains detector evidence between samples.
Lifecycle state	Active-event flag, start sample, event length, alarm-sample count, clear counter, clearance reason.	Implements `START`, `CLEAR`, and `GAP_TERMINATED` lifecycle transitions.
Event-summary state	Peak $\lvert z\rvert$, peak signed z, end signed z, peak residual, end residual, event direction, peak direction.	Stores compact event evidence for event-packet emission.
Context and suppression state	Gap flags, nearby-gap context, transient-spike confirmation state, quiet-reset state.	Prevents telemetry gaps and isolated point disturbances from contaminating event interpretation.

This static-state structure is one of the main differences between the offline reference and the embedded implementation. The offline implementation can retain rich intermediate outputs for plotting, event browsing, and post hoc analysis. The embedded implementation must instead retain only the state needed for causal decision-making and packet generation.

11.5. On-Board Prediction and Replay Modes

Several replay modes are used to isolate different parts of the embedded implementation. Detector-only modes test the decision layer using host-provided predictions. On-board prediction modes test whether the firmware can compute the expected temperature before residual scoring. Adaptive modes test whether the node-wise adaptive bias-correction path is active and causal on the MCU.

Table 11.4 summarizes the replay modes used by the prototype. The fixed on-board mode is used for the main embedded replay. The adaptive on-board mode is used to confirm that node-wise adaptive correction can run on the target hardware and reduce representative persistent residual bias under controlled mismatch replay. Verbose modes are useful for debugging and agreement inspection, but they are not used for the reported timing results because verbose serial output adds communication overhead.

Table 11.4: Host-to-STM32 replay modes used by the embedded prototype.

Mode	Host supplies	STM32 computes	Purpose
`S`	Observed temperature and host prediction.	Detector and event lifecycle only.	Detector-only replay with prediction held fixed by the host.
`Q`	Observed temperature and host prediction.	Detector, lifecycle, and verbose state output.	Debugging and per-sample agreement inspection for detector-only replay.
`T`	Observed temperature and orbital/context fields.	Fixed on-board predictor, residual evidence, detector, and lifecycle.	Main fixed on-board prediction replay mode.
`V`	Observed temperature and orbital/context fields.	Fixed on-board predictor, detector, and verbose state output.	Debugging and per-sample agreement inspection for fixed on-board prediction.
`A`	Observed temperature and orbital/context fields.	Node-wise adaptive on-board predictor, residual evidence, detector, and lifecycle.	Adaptive on-board prediction replay mode.
`W`	Observed temperature and orbital/context fields.	Node-wise adaptive predictor, detector, and verbose adaptive state output.	Debugging of adaptive bias, update gating, and agreement behaviour.
`P`	Orbital/context fields.	Predictor only.	Predictor diagnostic mode.

The adaptive implementation on the STM32 is limited to node-wise correction. This reflects the current embedded replay architecture, which processes one node stream at a time. Common-mode and hybrid correction require simultaneous residual information from multiple nodes, so they are evaluated only in the offline predictor-mismatch study. The embedded adaptive evidence in this thesis therefore concerns the fixed on-board predictor and the node-wise adaptive on-board predictor. This keeps the embedded claim aligned with the implemented firmware scope.

11.6. Host-to-Target Replay Workflow

The host-to-target replay workflow is shown from left to right in Figure 11.2. It provides deterministic embedded testing without requiring integration into the full spacecraft software stack. The host selects event-centred windows or full validation streams from the scored telemetry products, converts them into board rows, formats one UART input packet per sample, and streams those packets to the board. The STM32 processes the packets causally and returns event, diagnostic, and optional verbose packets. Host-side parsers then decode the returned packets for offline-to-embedded agreement analysis and replay-performance summaries.

The replay workflow also handles explicit telemetry gaps. If a replay window contains a time discontinuity, a synthetic gap row can be inserted so that the embedded detector receives a causal gap marker rather than silently processing the next available row as if no telemetry had been missing. This is necessary because the board sees only the samples sent to it; it cannot infer missing telemetry from rows that never arrive.

The resulting workflow is a controlled hardware replay. It preserves causal sample-by-sample execution on the MCU, while keeping event selection, scenario construction, and output comparison on the host. This makes the prototype suitable for offline-to-embedded agreement testing and resource-feasibility measurement, without treating the UART replay interface as a flight telemetry design.

11.7. Event and Diagnostic Packet Encoding

The embedded event packet is not a separate alert definition from the offline detector payload. It is the embedded encoding of the same detector-faithful event information: lifecycle phase, node, timing, residual magnitude, direction, onset evidence, audit evidence, and clear or gap-termination status. The difference is representation. The offline implementation can store richer Python event objects, while the embedded firmware emits fixed packet fields and encoded flags suitable for replay parsing, storage, or future downlink packaging.

Three output packet classes are used during replay. Event packets report lifecycle transitions. Diagnostic packets report interval-level timing and detector-state summaries. Verbose packets expose per-sample internal state for debugging and agreement inspection. Table 11.5 summarizes these packet classes.

The packet flow is shown in the centre and right-hand portions of Figure 11.2: the firmware emits `E`, `D`, and `V` packets, and the host parser decodes these packets before agreement and replay-performance analysis.

Table 11.5: Embedded packet types used during STM32 replay.

Packet type	Emitted when	Main fields	Thesis role
Input packet	Once per replay sample.	Sample index, node, observed temperature, prediction or orbital context, gap flag.	Provides causal sample input to the board during prototype replay.
Event packet	At `START`, `CLEAR`, or `GAP_TERMINATED` transitions.	Lifecycle phase, event ID, start sample, event length, detector flags, alarm count, peak residual evidence, direction, current sample.	Prototype encoding of the intended compact alert product.
Diagnostic packet	At configured diagnostic intervals.	Interval sample count, timing cycles, candidate count, active-event state, peak interval evidence, reason flags.	Supports timing, packet-size, and state-summary evaluation.
Verbose packet	Only in verbose replay modes.	Per-sample residual, normalized residual, cumulative evidence, flags, active state, prediction, adaptive bias, update gate.	Supports debugging and agreement inspection; not used for reported timing results.

The event packet preserves the onset-evidence and audit-evidence distinction defined in Section 10.6. Onset evidence records the detector evidence present when the event was declared, while audit evidence records relevant evidence accumulated across the event. Preserving both fields allows the embedded packet to retain the original declaration reason as well as later event context, such as a gap flag that occurs after event onset.

11.8. Offline-to-Embedded Verification Workflow

The embedded implementation is compared against the offline Python reference using event-level agreement rather than only sample-level traces. This is appropriate because the operational output of the detector is a bounded event lifecycle, not a continuous residual stream. The primary question is therefore whether the STM32 implementation preserves the same event-lifecycle behaviour under replay.

The right-hand side of Figure 11.2 summarizes the host-side verification path. The offline detector produces reference events, while the STM32 produces event packets. Host-side agreement scripts decode the embedded packets, match terminal events, compare lifecycle sequences, and compare event-summary fields within configured tolerances.

Lifecycle agreement is treated as the primary offline-to-embedded agreement metric. Exact equality of every event field is less suitable as the main criterion because embedded replay can differ slightly in event boundary timing, nearby-gap context, clear metadata, or audit flags. These differences may affect auxiliary fields without changing the event-lifecycle behaviour that matters for alerting. The agreement workflow therefore separates several layers of comparison:

1. lifecycle-sequence agreement, based on `START`, `CLEAR`, and `GAP_TERMINATED` transitions;
2. terminal-event matching, including missing and extra embedded terminal events;
3. numeric event-summary agreement, including event length, alarm samples, peak residual evidence, and end residual evidence;
4. direction agreement, including event direction and peak direction;
5. core evidence-flag agreement, separated from context-sensitive audit metadata.

This layered comparison supports a more meaningful implementation claim than bitwise packet equality alone. It asks whether the embedded implementation preserves the detector behaviour that matters for alerting, while still recording where numeric or metadata differences occur.

11.9. Timing, Memory, and Packet-Size Instrumentation

The embedded prototype includes instrumentation for timing, memory, and packet-size evaluation. These measurements are defined here so that the resource results in Chapter 12 can be interpreted consistently.

Timing is measured from diagnostic packets emitted by the firmware. Each diagnostic packet summarizes timing over a short interval, rather than reporting a raw timing record for every individual sample. Group-level mean timing is therefore computed as a sample-weighted mean across diagnostic intervals. This avoids giving equal weight to short and long replay windows. The worst observed processing time is taken from the largest per-sample timing value reported during replay. It is an observed replay maximum, not a formal worst-case execution-time proof.

Memory use is measured from the compiled firmware image. Whole-firmware flash and RAM are the primary footprint results. Object-level linker-map inspection can identify approximate contributions from detector, predictor, and main firmware objects, but whole-firmware size remains the more reliable implementation metric.

Packet sizes are measured from the encoded replay packets. Event-packet size is the most relevant alert-product result. Diagnostic and verbose packet sizes are useful for replay instrumentation, but should not be interpreted as mandatory nominal downlink products.

Table 11.6 summarizes the measurement sources and interpretation limits.

The instrumentation is interpreted conservatively. It is sufficient to assess whether the prototype fits within plausible compute-time, memory, and packet-size constraints. It does not replace integrated flight-software testing, bus-contention analysis, or flight-board power measurement.

Table 11.6: Embedded measurement sources and interpretation boundaries.

Metric	Measurement source	Interpretation boundary
Mean processing time	Sample-weighted diagnostic timing intervals.	Replay timing for the prototype firmware, not formal schedulability proof.
Worst observed processing time	Largest per-sample timing value reported during replay.	Observed worst replay path, not a formal WCET.
Flash and RAM	Compiled firmware ELF and linker-map artifacts.	Prototype firmware footprint, not a complete integrated flight-software build.
Event-packet size	Encoded event packet length.	Estimate of compact alert-product size.
Diagnostic and verbose packet size	Encoded diagnostic and verbose packet length.	Replay and debugging overhead; not necessarily nominal downlink content.
Compute-duty estimate	Measured processing time divided by the 15 s sample cadence.	Timing-derived estimate only; not direct current or energy measurement.

11.10. Embedded Predictor-Mismatch Replay

In addition to event-centred replay, the embedded tooling supports full validation-stream replay under selected predictor-mismatch scenarios. These replays are used to confirm whether the fixed and node-wise adaptive on-board prediction modes behave as expected on the STM32.

The mismatch is introduced by perturbing the observed temperature sent to the board while leaving the on-board predictor parameters and orbital context unchanged. From the detector's perspective, this creates a predictor-mismatch condition because the residual is still computed as observed temperature minus predicted temperature. The board is reset once per node, mode, and scenario so that independent node streams do not share detector or adaptive state. Within each node stream, however, the adaptive bias evolves causally across the replay.

This embedded mismatch replay has a narrower purpose than the offline predictor-validity-envelope study in Section 12.8. It does not establish a flight-ready adaptive correction policy. Instead, it confirms that the node-wise adaptive path is implemented on hardware, updates causally, and can reduce representative persistent residual bias under controlled replay.

11.11. Reusable Detector Core and Verification Infrastructure

The implementation boundary in Table 11.1 separates three parts of the prototype: the reusable detector core, the verification infrastructure used to exercise it, and the flight-integration work that remains outside this thesis. This section summarizes that distinction so that the embedded results in Chapter 12 are interpreted at the correct level.

The reusable component is the STM32 detector core. It implements the selected on-board prediction, residual scoring, evidence update, event lifecycle, gap handling, transient-spike suppression, node-wise adaptive correction, and compact event-packet generation using bounded causal state. This is the embedded AD component evaluated for timing, memory footprint, packet size, offline-to-embedded agreement, and adaptive replay behaviour.

The verification infrastructure is the replay workflow around the detector core. The UART replay pathway, host-side window selection, packet parsing, agreement scripts, and replay-performance summaries make the prototype testable and measurable, but they are not intended as flight software. In a flight implementation, the replay parser would be replaced by spacecraft telemetry ingestion, while the comparison and performance-analysis scripts would remain part of ground verification and development testing.

The remaining work is therefore flight integration rather than proof of basic MCU feasibility. This includes connecting the detector core to the spacecraft telemetry task, alert storage and queue management, downlink scheduling, mode-aware detector configuration, watchdog and task-scheduling analysis, flight-board current measurement, configuration management, and broader validation of adaptive correction. These boundaries are carried into Chapter 12 and Chapter 14 when interpreting the embedded feasibility results.

11.12. Chapter Summary

This chapter presented the STM32L4-class implementation and host-to-target verification workflow for the selected thermal AD pipeline. The embedded core implements on-board prediction, residual scoring, cumulative evidence, event-lifecycle management, telemetry-gap handling, transient-spike suppression, node-wise adaptive correction, and compact event-packet generation using bounded causal state.

The host-to-target replay workflow provides deterministic sample streaming, packet parsing, offline-to-embedded agreement checking, and timing, memory, and packet-size instrumentation. This workflow remains separate from the future flight telemetry interface.

The resulting prototype is suitable for measuring offline-to-embedded agreement, processing time, memory footprint, packet size, and node-wise adaptive replay behaviour. It does not constitute an integrated flight-software implementation. The measured embedded results are evaluated in Chapter 12.

12

Evaluation

This chapter evaluates the end-to-end Anomaly Detection (AD) prototype developed in the previous chapters. The objective is not only to report detector metrics, but to determine whether the prototype provides operational value for the Delfi Twin case study and, more broadly, for downlink-constrained small-satellite monitoring. The evaluation asks whether the pipeline can recover alert-worthy temperature events, suppress ordinary telemetry nuisances, produce bounded detector-faithful alerts, and execute within Microcontroller Unit (MCU)-class timing, memory, packet-size, and compute-duty constraints.

The evaluation is organized around 5 questions:

1. Does the detector recover alert-worthy telemetry-quality anomalies and thermal faults under matched-predictor assumptions?
2. Does it avoid turning benign telemetry nuisances into excessive alert events?
3. Does the event lifecycle behave appropriately around telemetry gaps, residual evidence that should be reset, and transient spikes?
4. Does the residual detector remain useful when the nominal predictor becomes biased or misaligned with the observed telemetry?
5. Can the complete residual-to-alert prototype, including prediction, residual scoring, event lifecycle logic, adaptive correction, and packet generation, run on STM32L4-class hardware within the required resource margins?

The chapter separates matched-predictor evaluation from predictor-mismatch robustness. Matched-predictor results test whether the residual-to-event pathway works when the expected-temperature model remains aligned with the telemetry. Predictor-mismatch results test how the same detector behaves when residual bias develops because the observed telemetry no longer matches the fixed nominal predictor.

Deployment-Oriented Workflow Steps

6. Tune and evaluate the selected pipeline
Benchmark candidate method families as end-to-end pipelines, including preprocessing and score-to-alert logic, using the protocol defined above.

7. Implement and verify under embedded constraints
Implement the selected pipeline on the target platform, demonstrate causal operation (no future samples), profile timing/memory, and validate robustness on the defined scenario suite.

12.1. Evaluation Evidence Layers and Claim Boundaries

The evaluation is organized as complementary evidence layers rather than a single performance score. This reflects the different questions the thesis must answer: whether labelled events are recovered, benign nuisances are suppressed, predictor mismatch affects reliability, the implementation runs causally on STM32L4-class hardware, and the workflow remains interpretable on real on-orbit telemetry.

The labelled synthetic benchmark provides the main quantitative event-level evaluation because it supplies controlled event truth: known event timing, affected channels, alert-worthiness, and recovery state. This enables consistent measurement of recall, Time to Awareness (TTA), false-event burden, nuisance rejection, and event-lifecycle behaviour. The other evidence layers broaden the evaluation beyond the matched synthetic case: predictor-mismatch scenarios test robustness to residual bias, STM32L4 replay tests causal embedded execution with bounded state and compact packets, and FUNcube windows test interpretability on real telemetry with thermal structure, gaps, and artifacts.

Table 12.1 summarizes the role and interpretation boundary of each evidence layer.

Table 12.1: Evaluation evidence hierarchy and claim boundaries for the deployment-oriented AD pipeline.

Evidence layer	Evaluation role	Supported claim	Claim boundary
Synthetic labelled benchmark	Controlled quantitative evaluation using known event labels.	Matched-predictor detector performance can be measured against known alert-worthy telemetry-quality anomalies and thermal-fault events.	Does not establish real flight fault statistics or real-data recall.
Decision-logic ablation	Component justification under fixed-predictor residuals.	The final event lifecycle is not reducible to simple thresholding; cumulative evidence, nuisance handling, and gap semantics affect event-level detector behaviour.	Does not prove that the selected configuration is globally optimal.
F1 adaptive-limits baseline	Simple causal rules/statistics comparator.	Causal local adaptive limits on observed temperature do not satisfy the recall, alert-burden, and event-usability trade-off for this deployment objective.	Does not reject all rule-based or statistical methods in general.
Predictor-mismatch robustness	Offline stress testing of residual detection under predictor bias or misalignment.	Fixed residual detection is limited by predictor validity; bounded adaptive correction can reduce mismatch-driven alert burden.	Does not establish a flight-ready adaptive correction policy.
Fixed embedded replay	On-target execution of the fixed on-board predictor and detector.	The fixed predictor, residual detector, event lifecycle, and event-packet pathway can run on STM32L4-class hardware with large timing and memory margins.	Does not prove full flight-software integration, real sensor-bus scheduling, or flight-board power.
Adaptive embedded replay	On-target execution of node-wise adaptive correction.	Node-wise adaptive prediction adds negligible runtime and memory overhead on the STM32 prototype.	Does not establish a flight-ready adaptive correction policy.
Embedded mismatch confirmation	Targeted hardware replay under controlled local Batt mismatch.	The embedded adaptive predictor can causally correct persistent and gradually varying local predictor mismatch on hardware.	Does not replace the broader offline mismatch study across nodes, scenarios, and fault families.
FUNcube-1 stress tests	Qualitative real-telemetry interpretability tests.	The workflow can be exercised on real on-orbit telemetry with thermal structure, gaps, and artifacts, producing inspectable residual, evidence, and event-state traces.	Does not provide labelled recall, precision, or confirmed fault detection.
Operational readiness synthesis	Engineering judgement across all evidence layers.	The prototype is suitable for continued engineering development and hardware-in-the-loop testing.	Does not establish autonomous flight readiness or beacon-level operational trust.

The thesis claims are therefore additive but bounded. The synthetic benchmark supports controlled event-level performance claims; the ablation and baseline comparisons justify the selected residual-to-alert design; the predictor-mismatch studies test robustness when residual bias develops; the STM32L4 replay supports embedded feasibility claims; and the FUNcube windows test whether the workflow remains interpretable on real on-orbit telemetry. Together, these layers support a deployment-oriented prototype claim: the pipeline is suitable for continued engineering development and hardware-in-the-loop testing. Flight performance, real-fault statistics, complete fault coverage, autonomous recovery, and flight-ready operational trust remain beyond the claim scope of this thesis.

12.2. Role and Interpretation of the Synthetic Benchmark

The primary quantitative detector evaluation in this chapter uses the labelled synthetic evaluation split introduced in Chapter 7. This benchmark provides controlled event truth for the residual-to-alert pipeline: event onset, event offset, affected node, anomaly class, alert-worthiness, and recovery state are known by construction. These labels make it possible to measure alert-worthy recall, TTA, false-event burden, nuisance sensitivity, event fragmentation, and gap-aware lifecycle behaviour in a reproducible way.

This role is complementary to real-telemetry evaluation. Real spacecraft telemetry provides valuable realism, but it rarely provides independent event-level labels for the thermal fault and telemetry-quality classes considered here. For the available Delfi telemetry specifically, the continuity check in Section 4.3 showed that the ground-received samples are too sparse and gap-dominated for continuous detector replay, with the densest 95 min window containing only 12 samples. The synthetic benchmark therefore supplies the controlled event truth needed for quantitative evaluation, while real telemetry is used separately for qualitative stress testing where labels are unavailable.

The benchmark is used as a controlled evaluation substrate rather than as a flight-certified thermal simulator. Its value is that the nominal baseline, observation-layer effects, telemetry-quality anomalies, and thermal-fault injections are known, repeatable, and traceable to the telemetry scope and fault families defined earlier in the thesis. This allows the evaluation to test whether the detector forms bounded events, suppresses benign nuisances, handles gaps, and reports detector-faithful alert evidence under controlled conditions, without treating the resulting performance metrics as predicted flight performance.

To support fair interpretation, the evaluation follows a fixed split discipline. Nominal data are used to construct the expected-temperature predictor and nominal residual statistics. The validation split is used to tune detector settings and select comparison configurations. The evaluation split is used only after the detector configuration is frozen. The F1 adaptive-limits baseline is treated the same way: its (W, k) parameters are swept on validation, representative Pareto configurations are selected from that validation trade-off, and those configurations are frozen before evaluation.

The synthetic benchmark therefore answers a specific and useful question: whether the proposed prediction, residual-scoring, and event-lifecycle pathway behaves correctly under controlled labelled conditions. Its results are interpreted alongside predictor-mismatch stress tests, embedded replay, and FUNcube-1 real-telemetry stress tests. Together, these layers support prototype-level claims, while real-flight recall, real false-alert rates, complete fault coverage, and autonomous deployment readiness remain outside the benchmark claim.

12.3. Evaluation Protocol and Metrics

The quantitative evaluation follows the train, validation, and evaluation split structure defined in Chapter 7. The training split is used to fit the expected-temperature predictor and estimate nominal residual statistics. The validation split is used to select detector settings, tune baselines, and assess robustness during development. The evaluation split is reserved for final performance reporting after all configurations are fixed.

The metrics used throughout the chapter are summarized in Table 12.2 and grouped by the evidence layer they support. Not all metrics apply to all settings. For example, FUNcube-1 stress tests are not assigned precision or recall because independent fault labels are unavailable.

Table 12.2: Main metrics used in the evaluation chapter and their interpretation.

Metric	Definition	Interpretation
Alert-worthy recall	Fraction of telemetry-quality and thermal-fault truth events overlapped by at least one detector event.	Measures controlled event coverage on labelled synthetic data.
Detector events	Number of bounded events produced by the detector.	Measures alert-stream burden and event fragmentation.
False detector events	Detector events that do not overlap alert-worthy truth events.	Measures non-alert-worthy event burden under the synthetic label scheme.
False events/-day	False detector events normalized by the scored evaluation duration.	Measures operational false-alert burden for comparing configurations.
Time-to-awareness	Time from labelled truth-event onset to the first overlapping detector event.	Measures detector declaration delay, not downlink or ground-operator awareness time.
Short events (≤2) samples	Detector events lasting no more than two telemetry samples.	Measures short-event chatter and threshold-fragment behaviour.
Nuisance escalations	Detector events associated with benign nuisance behaviour rather than alert-worthy truth events.	Measures whether benign telemetry nuisances are escalated into detector events.
Clear/restart cycles	Repeated detector clear and restart behaviour within related truth or event intervals.	Measures event fragmentation and lifecycle instability.
Events active during missing telemetry	Number of detector events that remain active while observations are unavailable.	Measures whether events remain open across intervals where telemetry is unavailable.
False-alert occupancy	Percentage of monitored node-time falsely spent in alert.	Measures sustained false-alert state under predictor mismatch.
Processing time/sample	Measured STM32 processing time for replayed telemetry samples.	Measures real-time feasibility relative to the 15 s telemetry cadence.
Memory footprint	Firmware flash and RAM use from STM32 builds.	Measures embedded resource feasibility and adaptive overhead.
Tracking error	Adaptive bias minus imposed target mismatch during embedded mismatch replay.	Measures whether embedded adaptation recentres controlled local predictor mismatch.

The main offline evaluation operates at the event level. Truth labels define intervals containing injected anomalies or nuisances, and detector outputs are bounded event intervals. A detector event is counted as correct when it overlaps an alert-worthy truth interval for the same node. Alert-worthy recall is computed over telemetry-quality and thermal-fault events; benign nuisances are excluded from recall but contribute to false-burden and nuisance-rejection metrics. False-event rates are normalized by the scored evaluation duration, using 730.0 day for ablation and baseline comparisons. Subset analyses, such as predictor-mismatch replay, report event burden over the specific evaluation window.

Predictor-mismatch scenarios are treated as validation-stage stress tests rather than additional fault classes. Clean mismatch cases test whether residual bias creates false alerts on nominal data, while injected mismatch cases test whether true events remain detectable under shifted baselines. TTA is reported as detector declaration delay, from truth-event onset to the first overlapping detector event. It excludes contact gaps, downlink delay, and ground processing, and therefore reflects on-board detection latency only during observable telemetry.

Embedded evaluation is performed through event-centred replay on an STM32L476RG prototype. Fixed-mode replay evaluates residual detection with a static on-board predictor, while adaptive replay adds a causal bias-correction layer before scoring. Embedded performance is reported as processing time per sample relative to the 15 s telemetry cadence and as memory footprint from firmware builds with and without adaptation. The remainder of the chapter follows the evidence hierarchy in Table 12.1: matched-predictor synthetic results, ablations and baseline comparison, predictor-mismatch robustness, and STM32 replay validation.

12.4. Matched-Predictor Synthetic Benchmark Results

This section evaluates the offline Python reference detector on the evaluation split after all detector settings were fixed. The purpose is to isolate the residual-to-alert pathway under matched-predictor conditions: the expected-temperature model remains aligned with the observed telemetry, so detector events are driven primarily by injected telemetry-quality anomalies, thermal faults, benign nuisances, and nominal residual variation rather than by predictor mismatch. This provides the controlled event-recovery baseline before later robustness tests introduce predictor bias or real-telemetry stress cases.

Under matched-predictor conditions, the detector recovered all alert-worthy truth events. The evaluation split contained 36 alert-worthy events, consisting of telemetry-quality anomalies and thermal faults, and all 36 were detected. The same split contained 207 benign nuisance labels, which are no-alert-expected robustness cases rather than positive detection targets. Most benign nuisance intervals were suppressed: 183/207 had no detector overlap. The remaining 24/207 benign-label overlaps were concentrated into 13 bounded detector events because several nuisance labels occurred close together or within the same detector interval.

The resulting alert stream therefore achieved complete alert-worthy recall while keeping the nuisance-associated event burden low. Across the 730.0 day evaluation split, the 13 non-alert-worthy detector events correspond to a normalized burden of $0.0184\,\mathrm{d}^{-1}$. This result supports the matched-predictor claim that the selected residual evidence and event-lifecycle logic recover alert-worthy events while keeping ordinary nuisance responses bounded and infrequent. The results are summarized in Table 12.3.

Table 12.3: Matched-predictor event-detection outcome on the evaluation split.

Metric	Value	Interpretation
Evaluation duration	730.0 d	Evaluation split used after detector configuration was fixed
Total truth labels	243	Benign nuisance, telemetry-quality, and thermal-fault labels
Alert-worthy truth events	36	Telemetry-quality anomalies and thermal-fault events
Alert-worthy truth events detected	36 / 36	Complete alert-worthy recall under matched-predictor conditions
Benign nuisance labels	207	No-alert-expected robustness cases
Benign nuisance labels without detector overlap	183 / 207	Most benign nuisance intervals were suppressed
Benign nuisance labels with detector overlap	24 / 207	Benign labels that coincided with detector output
Detector events	51	Bounded detector events produced over the evaluation split
Detector events overlapping alert-worthy truth	38	Alert-worthy detector events, including two fragmented heater-control cases
Nuisance-associated detector events	13	Bounded detector events associated only with benign nuisance intervals
Nuisance-associated event rate	$0.0184\,\mathrm{d}^{-1}$	Approximately one non-alert-worthy detector event per 54 d; comparison metric, not a predicted flight rate

Two counts are useful here because they answer different questions. The first is truth-event recall: did the detector find each labelled alert-worthy event? This is the main matched-predictor result, because the priority is to avoid missing telemetry-quality anomalies and thermal faults. On this measure, the detector recovered all 36/36 alert-worthy truth events.

The second count is detector-event precision: of the bounded events emitted by the detector, how many overlapped alert-worthy truth? This describes the purity of the alert stream rather than the detector's ability to recover labelled faults. The detector produced 38 true detector events for 36 alert-worthy truth events because two heater-control-failure events were split into separate detector events when telemetry gaps interrupted observability.

Expressed as detector-event precision, 38/51 bounded detector events overlapped alert-worthy truth events. This value is useful for describing the emitted alert stream, but it should not be treated as the sole measure of matched-predictor performance. The more operationally informative result is the combination of complete alert-worthy recall and low nuisance-associated event burden across the full evaluation duration.

The normalized non-alert-worthy event burden is reported as a configuration-comparison metric over the scored benchmark duration. It should not be interpreted as a predicted flight false-alert rate because the event density, nuisance density, and alert-worthy event frequency are defined by the synthetic scenario design. Under matched-predictor conditions, the detector recovered all 36 alert-worthy truth events while producing only 13 bounded non-alert-worthy detector events despite exposure to 207 benign nuisance labels. This corresponds to $0.0184\,\mathrm{d}^{-1}$, or approximately one non-alert-worthy detector event per 54 d in the synthetic evaluation split.

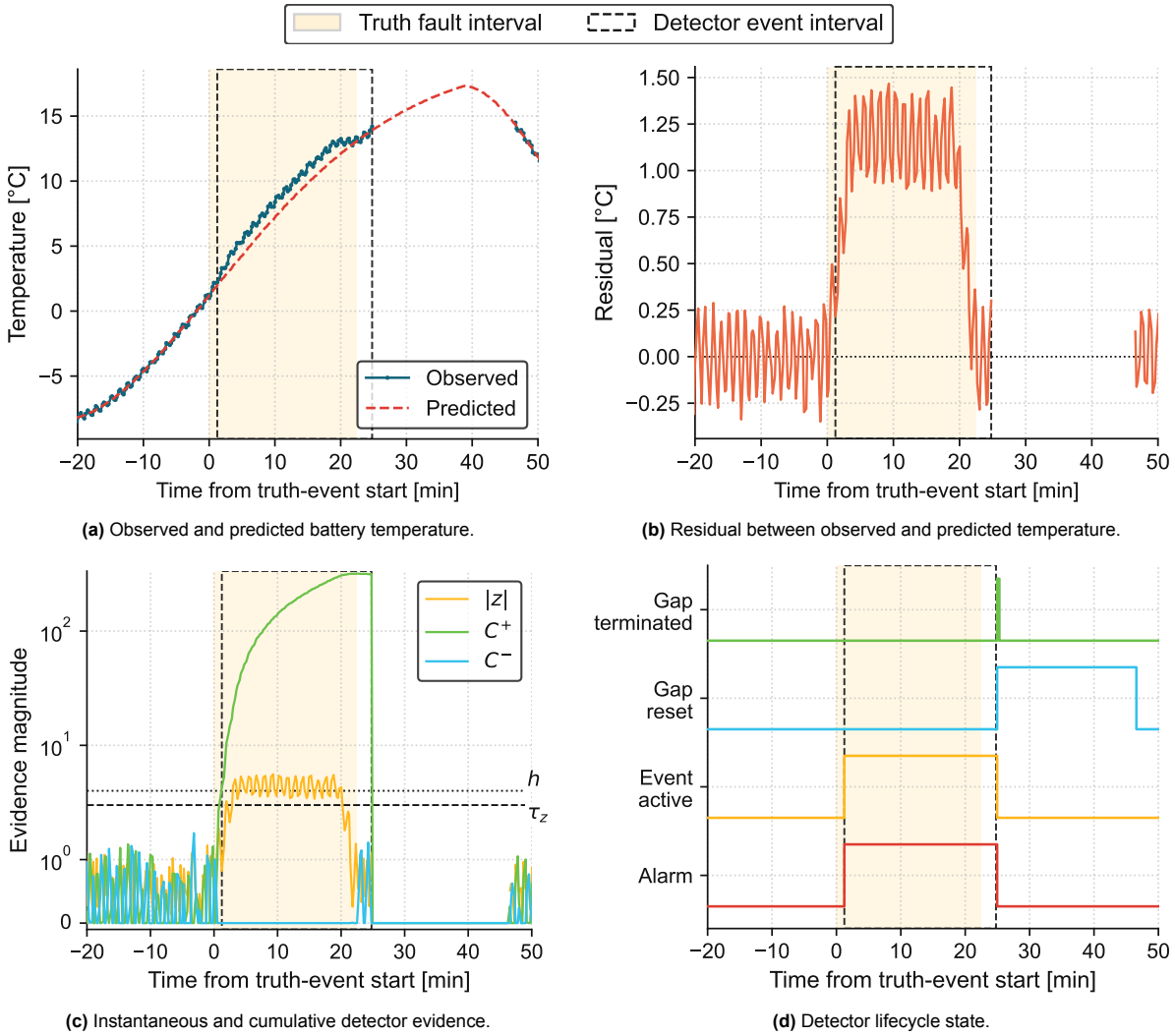
The category-level result shows complete recall for both telemetry-quality anomalies and thermal faults. The aggregate thermal-fault TTA, however, is less informative on its own because it is dominated by heater-control failures. These are gradual thermal deviations and, in this evaluation, were also affected by telemetry gaps. The subtype-level result in Table 12.4 therefore gives a clearer view of detector behaviour across event types. Abrupt calibration steps and pinned values are detected quickly because they produce strong telemetry-quality deviations. Battery thermal runaway is detected quickly because the residual grows rapidly, while self-heating is also detected on a short timescale.

Table 12.4: Detection performance by alert-worthy subtype. Heater-control failures are detected reliably but later than the other fault types because they are gradual events and were affected by telemetry gaps.

Category	Subtype	Truth	Hit	Recall [%]	Median TTA [s]	P95 TTA [s]	Detector events per truth event
TQA	Abrupt calibration step	12	12	100	31.6	94.9	1.00
TQA	Pinned values	18	18	100	58.0	99.7	1.00
TF	Battery thermal runaway	2	2	100	31.6	31.6	1.00
TF	Heater control failure	2	2	100	796.3	839.0	2.00
TF	Self-heating	2	2	100	79.1	93.3	1.00

A representative true-positive thermal-fault example is shown in Figure 12.1. The figure shows the observed and predicted temperatures, residual evidence, instantaneous and cumulative evidence, and event-state transition. Its purpose is to show how a physical thermal deviation is converted into detector evidence and then into a bounded detector-faithful alert event.

This example also illustrates the added value relative to conventional upper/lower temperature-limit monitoring. Simple limit checking, introduced in Subsection 2.9.1, remains necessary for hard protection, but it asks a different question from the residual detector. A fixed maximum or minimum temperature limit asks whether the measured temperature has left an allowable range. The residual detector asks whether the measured temperature is abnormal for its expected thermal context. In this representative event, the relevant evidence is therefore not only the absolute battery temperature, but the sustained positive deviation between observed and predicted temperature. Without residual evidence and event-lifecycle logic, this type of contextual deviation may remain visible only after ground inspection of the downlinked telemetry.



(a) Observed and predicted battery temperature.

(b) Residual between observed and predicted temperature.

(c) Instantaneous and cumulative detector evidence.

(d) Detector lifecycle state.

Figure 12.1: Representative self-heating fault detection showing residual growth, cumulative evidence, event activation, and event clearing.

The detector summarizes this evidence as a compact event packet. Table 12.5 shows a decoded example for the same representative Batt self-heating truth event. The packet reports the affected node, event timing, residual direction and magnitude, triggering evidence, clearance reason, and continuity caveats. It therefore provides detector-faithful evidence for ground review without requiring the full diagnostic plot to be downlinked.

This decoded packet illustrates the intended explainability level of the prototype. It reports detector-faithful evidence: the Batt node was hotter than expected, instantaneous and cumulative evidence contributed to event declaration, the event persisted long enough to be opened, and it later cleared by hysteresis. The packet therefore provides compact evidence for ground review without converting the alert into an on-board root-cause diagnosis.

The matched-predictor result establishes that the selected residual-to-alert pathway works under the nominal assumptions of the benchmark. Its deployment robustness over longer mission timescales depends on whether the expected-temperature predictor remains aligned with observed spacecraft telemetry. This predictor-validity question is evaluated explicitly in Section 12.8.

Table 12.5: Decoded detector-faithful alert packet for the representative Batt self-heating event in Figure 12.1.

Operator question	Decoded alert-packet evidence
What happened?	Hotter-than-predicted residual event.
Where?	Batt temperature node.
How severe?	Peak $z = +5.60$, peak $\lvert z\rvert = 5.60$, peak residual $= +1.47\,°C$.
How long?	23.2 min; 94 event samples, including 94 alarm samples.
Why did it alert?	Positive instantaneous threshold evidence, positive cumulative evidence, and persistence rule met.
Is it ongoing?	No. The event terminated by data gap; end residual $= +0.30\,°C$, end $z = +1.15$.
Any caution?	Event was terminated because telemetry observability was lost; recovery was not directly observed.
On-board claim boundary	Detector-declared residual anomaly; not root-cause diagnosis.

12.5. Nuisance, Gap, and Event-Lifecycle Behaviour

This section evaluates how the detector handles benign telemetry nuisances and gaps under matched-predictor conditions. These cases are not positive detection targets. Instead, they test whether ordinary telemetry imperfections are rejected, reset, or bounded before they become unnecessary alert events.

As shown in Table 12.6, the detector overlapped 24 of 207 benign nuisance labels. These overlaps came primarily from isolated spikes and ordinary noise windows, while small quantization produced few overlaps. At the detector-output level, however, the 24 nuisance-label overlaps were consolidated into 13 bounded non-alert-worthy detector events because several labelled nuisance intervals occurred close together or within the same detector event.

The nuisance-label overlap count identifies which benign perturbations stress the residual evidence and lifecycle logic. The bounded detector-event count is the more operationally relevant alert-burden measure, because it represents the number of non-alert-worthy events that would enter the alert stream. This result shows that benign nuisances were not ignored perfectly, but their effect was limited to a small number of bounded detector events over the full evaluation split.

Table 12.6: Benign truth events overlapping detector alerts. These values describe nuisance sensitivity at the truth-event level, not the number of false detector events.

Benign subtype	Truth events	Overlapped by detector event	Trigger rate
Isolated spikes	123	17	13.8
Ordinary noise	48	6	12.5
Small quantization	36	1	2.8
Benign total	207	24	11.6

False-alert detector events were shorter and lower magnitude than true-alert events. The median false-alert duration was much shorter than the median true-alert duration, and the median peak $\left|z_{j,t}\right|$ was also lower. These observations derive from the results presented in Table 12.7. This supports the interpretation that most false alerts are bounded nuisance responses rather than persistent detector failure. From an operator perspective, the matched-predictor alert burden is low: 13 false detector events over 730.0 d.

Table 12.7: True-alert and false-alert detector-event properties.

Detector-event class	Count	Median duration [s]	Mean duration [s]	Max duration [s]	Median peak $\left\lvert z_{j,t}\right\rvert$	Max peak $\left\lvert z_{j,t}\right\rvert$
True alert	38	1344.7	1815.5	4461.3	14.5	461.6
False alert	13	179.3	200.4	632.8	3.20	51.1

Telemetry gaps are handled separately because they represent loss of observability. If telemetry is unavailable during an active event, the detector cannot determine whether the anomaly recovered or persisted during the unobserved interval. Active events are therefore terminated as `GAP_TERMINATED`, rather than bridged across the gap or cleared as normal. Figure 12.2 shows a heater-control-failure case in which the detector declares thermal-fault events during observable segments and terminates active events when telemetry observability is lost.

This behaviour supports the intended lifecycle semantics: the detector reports bounded events during observable telemetry, while preserving the fact that recovery was not directly observed across gaps.

12.6. Decision-Logic Ablation Study

A decision-logic ablation study was performed on the synthetic evaluation split to test what the full alert lifecycle adds beyond simple residual thresholding. All variants used the same expected-temperature predictor, residuals, normalized residuals, monitored nodes, truth labels, and event-matching procedure. Only the decision-layer configuration was changed. The comparison therefore isolates the effect of persistence, cumulative evidence, nuisance handling, and lifecycle stabilization on alert-worthy recall, false-event burden, nuisance escalation, short-event chatter, and gap handling.

Table 12.8 summarizes the main ablation results. The threshold-only variant used residual-threshold evidence with minimal event grouping and the common gap-termination rule. This provides a simple reference case for comparing threshold evidence alone against the complete residual-to-alert lifecycle. The rows are ordered from a simple threshold-only reference, through two targeted removals from the selected lifecycle, to the final detector configuration.

Table 12.8: Decision-logic ablation results on the synthetic evaluation split.

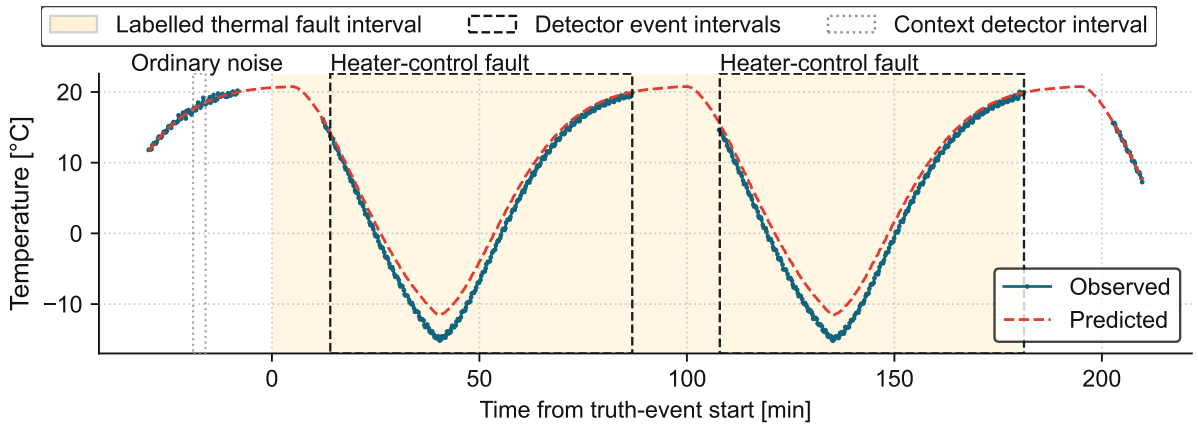
Configuration	Recall [%]	Detector events	False events/day	Nuisance escalations	Events ≤ 2 samples	Clear/restart cycles	P95 TTA [s]
Threshold-only reference	86.1	233	0.1630	119	150	83	548.4
Final without CUSUM	80.6	35	0.0000	0	0	6	601.2
Final without nuisance handling	100.0	94	0.0767	56	0	2	282.1
Final detector	100.0	51	0.0178	13	0	2	282.1

The ablation shows that the final detector is not reducible to simple thresholding. The threshold-only variant has lower alert-worthy recall and produces far more detector events, nuisance escalations, short events, and clear/restart cycles. It produces 233 detector events, 119 nuisance escalations, and 150 short events ((≤2) samples), compared with 51, 13, and none for the final detector. This supports a central design choice of the thesis: residual threshold crossings should become candidate evidence for an event lifecycle, not operational alerts by themselves.

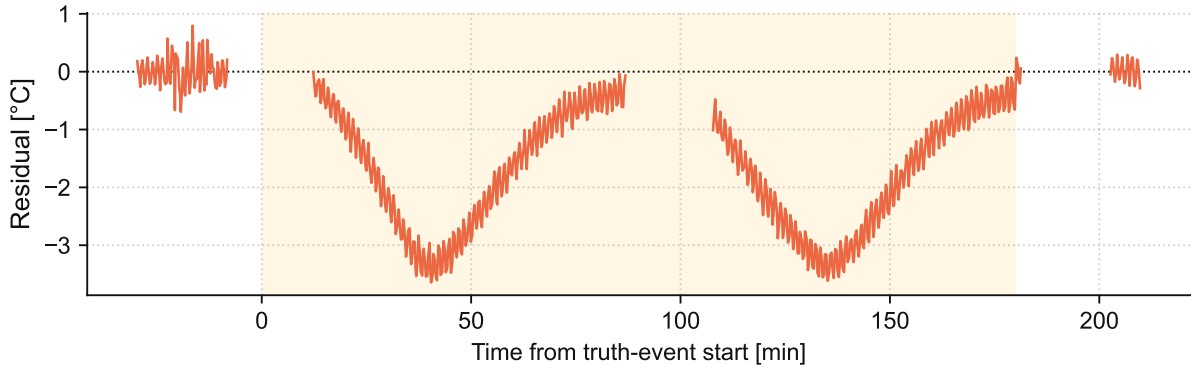
The component ablations clarify the role of individual decision elements. Removing CUSUM eliminates false events, but reduces recall from 100.0 % to 80.6 % and increases P95 TTA from 282.1 s to 601.2 s. Cumulative evidence is therefore needed to recover persistent low-amplitude deviations, despite some additional false-event risk.

Removing nuisance handling preserves recall and P95 TTA, but increases nuisance-related alert burden from 13 to 56 non-alert-worthy detector events. This corresponds to a change from about one non-alert-worthy event per 56 d to about one per 13 d. Nuisance handling therefore mainly improves alert-stream stability and operational usability, rather than sensitivity.

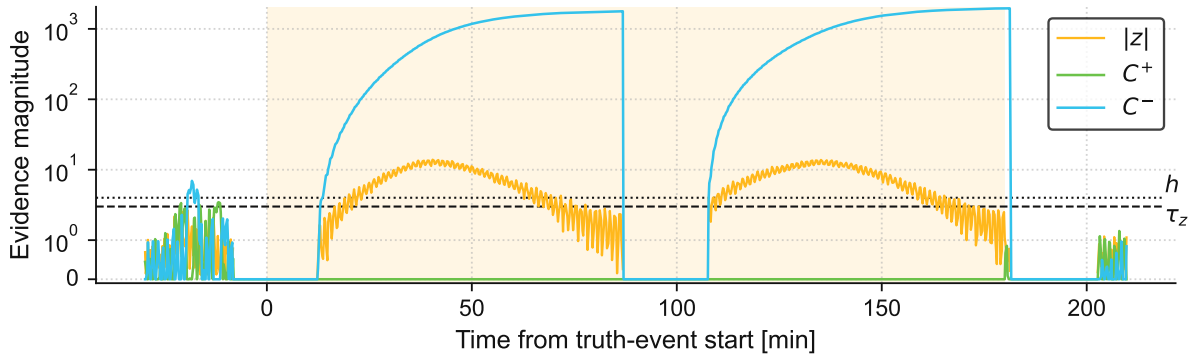
Gap termination is an event-semantics mechanism rather than a conventional performance feature. Without it, 6 events remain active during missing telemetry, corresponding to 126.6 s of active-event state without observability. The final detector instead produces 6 `GAP_TERMINATED` events and leaves no events active during missing telemetry. Gap termination therefore prevents event continuity from being carried across unobserved intervals; it marks loss of observation, not physical recovery.



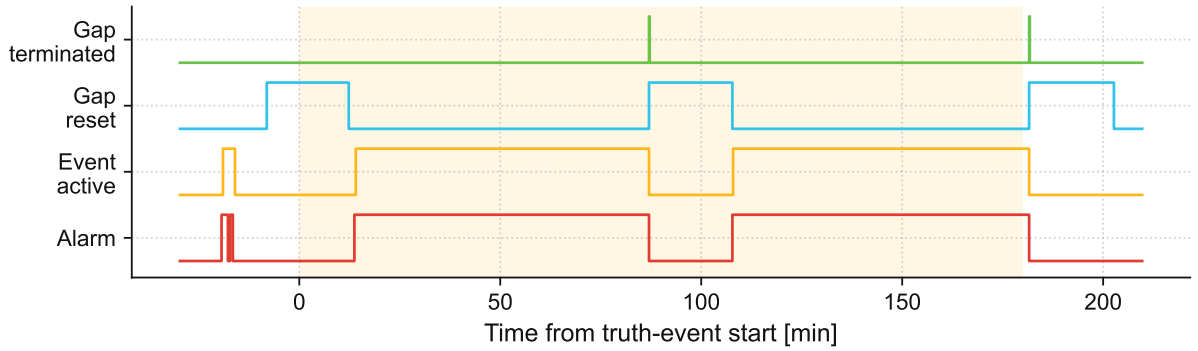
(a) Observed and predicted MCU temperature.



(b) Temperature residual.



(c) Instantaneous and cumulative detector evidence.



(d) Detector lifecycle state.

Figure 12.2: Gap-aware detector lifecycle during a heater-control fault, showing event termination at telemetry gaps rather than bridging unobserved intervals.

The persistence and hysteresis ablation is not used as a primary design justification. In this evaluation split, removing persistence and hysteresis produced similar aggregate results to the final detector, with 100.0% recall, 48 detector events, 0.0137 false events/day, and P95 time-to-awareness of 263.7 s. These rules are therefore retained as conservative lifecycle safeguards for stability and event semantics, but this ablation does not identify them as dominant contributors to the aggregate held-out performance.

Overall, the ablation study supports the selected decision logic as an operational alert-forming layer rather than a purely statistical threshold rule. The main performance gains come from combining cumulative evidence with nuisance handling and event lifecycle semantics. Threshold-only detection is too fragmented and nuisance-sensitive, while removing CUSUM makes the detector too conservative. Gap termination contributes primarily to observability honesty rather than aggregate recall or false-alert metrics. These results justify the final detector configuration used for the subsequent baseline comparison, predictor-mismatch analysis, and embedded replay.

12.7. Method-Family Baseline: F1 Adaptive Limits

The ablation above tested what the selected alert lifecycle adds once normalized residual evidence is available. The F1 adaptive-limits baseline tests a more basic question: whether a simple rules, limits, and statistics method can satisfy the deployment objective without the expected-temperature predictor and residual-to-alert pathway.

For this purpose, a causal adaptive-limits baseline was implemented directly on the observed calibrated temperature streams. For each node, the baseline estimated a rolling mean and standard deviation from past samples only,

$$\mu_t = \text{mean}\left(y_{t-W}, \ldots, y_{t-1}\right), \qquad \sigma_t = \text{std}\left(y_{t-W}, \ldots, y_{t-1}\right). \tag{12.1}$$

where y_t is the observed temperature sample and W is the causal rolling-window length. A sample was flagged when

$$|y_t - \mu_t| > k\sigma_t, \tag{12.2}$$

where k is the adaptive-limit multiplier. Contiguous limit violations were grouped into detector events.

The baseline does not use expected-temperature prediction, residual normalization, CUSUM, persistence, hysteresis, nuisance handling, or the final event lifecycle. It therefore represents a simple onboard-compatible rules/statistics comparator rather than a complete residual-to-alert pipeline.

The F1 parameters W and k were swept on the validation split. Figure 12.3 shows the resulting recall versus false-alert trade-off. Representative Pareto-efficient configurations were selected from this validation sweep and frozen before evaluation, following the same validation-to-evaluation discipline used for the final detector.

Table 12.9 reports the evaluation performance of the frozen F1 configurations and compares them with the final residual-to-alert detector. The low-burden F1 configuration produced few false events, but detected only 41.7 % of alert-worthy events. Increasing sensitivity improved recall, but made the alert stream unusable: the higher-recall F1 configuration reached 91.7 % recall, but produced 34 516 detector events, 47.2 false events per day, and 34 514 short events lasting no more than two samples.

Table 12.9: Held-out evaluation comparison between validation-selected F1 adaptive-limits configurations and the final residual-to-event detector. F1 configurations were selected from the validation Pareto sweep and frozen before evaluation.

Method / configuration	Recall [%]	Detector events	False events per day	Short events ≤ 2 samples	P95 TTA [s]
F1 low-burden Pareto, $W = 68, k = 6.0$	41.7	18	0.0041	17	1376.4
F1 higher-recall Pareto, $W = 8, k = 3.0$	91.7	34516	47.2	34514	1860.5
Final residual-to-event detector	100.0	51	0.0178	0	282.1

In contrast, the final residual-to-alert detector achieved complete alert-worthy recall with 51 detector events, no short-event chatter, and a P95 TTA of 282.1 s. This comparison shows that adaptive limits on observed temperature alone do not provide a useful operating point for this deployment objective: low-burden settings miss many alert-worthy events, while higher-recall settings produce excessive short-event output.

This result should not be interpreted as a rejection of all rule-based or statistical AD methods. It evaluates a representative causal local-limit baseline from the rules, limits, and simple-statistics family. Limit checking remains meaningful for hard protection and is consistent with conventional spacecraft telemetry monitoring and Delfi-PQ heritage Fault Detection, Isolation and Recovery (FDIR) practice [27, 54]. For the residual-to-alert objective in this thesis, however, the comparison supports the use of expected-temperature prediction, residual scoring, cumulative evidence, and event-lifecycle logic to form bounded alerts with acceptable event recovery and alert burden.

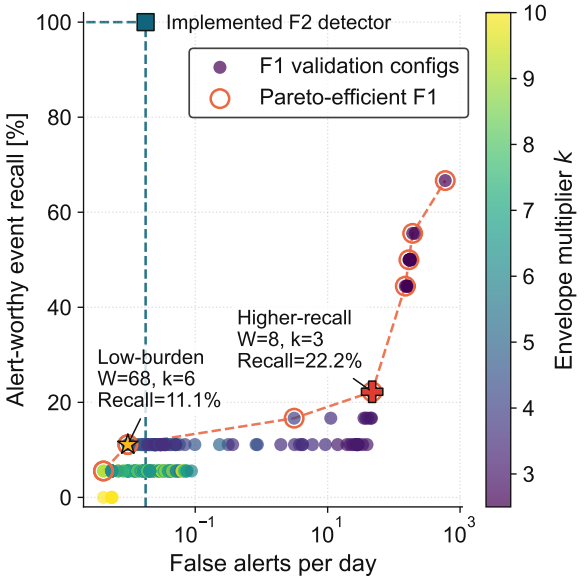


Figure 12.3: Validation recall versus false-alert burden for the causal F1 adaptive-limits baseline.

12.8. Predictor Validity Envelope and Adaptive Correction

The matched-predictor results show that the residual-to-alert detector can recover alert-worthy events when the expected-temperature model remains aligned with the telemetry. For deployment, this alignment is part of the detector validity question. A residual detector scores the difference between observed and expected temperature; if the expected-temperature model becomes biased or misaligned with the current operating context, residual evidence may reflect predictor mismatch rather than abnormal spacecraft behaviour.

This section evaluates the predictor-validity envelope of the fixed residual detector. The objective is to determine when the fixed predictor remains sufficient, when persistent residual bias creates excessive alert burden, and whether the bounded adaptive correction layer defined in Section 9.6 can reduce mismatch-driven alerts without suppressing alert-worthy event evidence.

This issue is especially important for the CUSUM branch. CUSUM is useful because it accumulates small sustained residual shifts, allowing gradual faults to be detected before they produce large instantaneous deviations. The same property also makes the detector sensitive to persistent predictor bias: a small offset that remains present for many samples can accumulate as detector evidence even when no bounded thermal fault has occurred.

The predictor-mismatch scenarios apply controlled temperature offsets to the validation data while keeping the predictor and detector configuration fixed. They are not treated as additional physical fault labels. Instead, they emulate residual-baseline shifts caused by effects such as calibration offset, thermo-optical ageing, thermal-interface change, or operating-context mismatch. Clean mismatch cases measure false-alert burden when no labelled anomaly is present. Injected mismatch cases test whether alert-worthy telemetry-quality and thermal-fault events remain detectable after the residual baseline has shifted. The scenarios are summarized in Table 12.10.

Table 12.10: Predictor-mismatch robustness scenarios in the validation analysis. Clean cases measure mismatch-induced nuisance alerting, while injected cases assess whether alert-worthy events remain detectable under shifted residual baselines.

Scenario	Temperature mismatch	Purpose
Baseline	No imposed mismatch	Matched-predictor reference case
Global $+0.5\,°C$	All nodes shifted by $+0.5\,°C$	Tests sensitivity to common residual bias
Ageing ramp	Panels ramp from $+[0..2.0]°C$ Batt/MCU ramp from $+[0..1.3]°C$	Tests physically motivated external/internal mission-ageing mismatch
Battery $+0.5\,°C$	Batt shifted by $+0.5\,°C$	Tests local offset behaviour and fault absorption risk

Under these validation-stage mismatch scenarios, the fixed-predictor detector becomes alert-burden limited. A global (+0.5 °C) mismatch causes approximately 93 false detector events per day, and the ageing-ramp scenario produces a similar burden. A battery-local (+0.5 °C) offset causes fewer false events, but still produces a persistent false-alert state. These results show that matched-predictor recall is not sufficient on its own: the detector can recover true events while also producing an alert stream dominated by predictor mismatch.

This behaviour should not be interpreted as a failure of CUSUM or of the event lifecycle. The detector is doing what it was designed to do: accumulating sustained residual evidence. The deployment issue is that the residual has lost its intended interpretation because observed temperature is no longer being compared against a valid expected-temperature estimate. The mismatch analysis therefore evaluates not just detector sensitivity, but the conditions under which residual evidence remains meaningful.

Adaptive correction is evaluated as a bounded residual-bias correction layer on top of the fixed physically shaped predictor. It is not a new black-box forecaster. The fixed base model remains inspectable, while the correction term attempts to reduce sustained predictor mismatch. The common-mode, node-wise, and hybrid bias-correction structures are defined in Section 9.6; this section uses them only to compare predictor-validity behaviour under mismatch.

All adaptive modes retained complete alert-worthy recall in the tested injected mismatch cases. Recall alone, however, is not enough for deployment assessment under predictor mismatch, because a detector can retain recall while also spending long periods in false alert. For this reason, false detector events and false-alert occupancy are interpreted alongside recall. False-alert occupancy is especially useful here because it captures persistent false-alert states that event counts alone can understate.

Common-mode correction handles true global mismatch well, but is less effective when the mismatch is node-dependent. Node-wise and hybrid correction reduce false-alert occupancy to near-baseline levels across the tested mismatch cases while retaining complete alert-worthy recall. The performance of each adaptive mode is presented in Table 12.11. Darker cells indicate higher false-alert occupancy; the colour scale is only a visual guide, and the printed values provide the quantitative comparison.

Table 12.11: False-alert occupancy under predictor mismatch. Values give the percentage of monitored node-time falsely spent in alert across the six monitored temperature nodes. Alert-worthy validation recall was 1.00 for all modes and scenarios.

Scenario	Fixed	Common-mode	Node-wise	Hybrid
Baseline	0.01	0.01	0.02	0.02
Global $+0.5\,°C$	**76.48**	0.02	0.02	0.02
Ageing ramp	**70.52**	**47.95**	0.02	0.02
Battery $+0.5\,°C$	**12.85**	**30.53**	0.02	0.02

The full recall, false-event-rate, and node-time alert-burden values underlying Table 12.11 are provided in Table D.1.

The battery-local case illustrates why shared correction is not sufficient for local mismatch. A common correction cannot isolate the Batt residual offset from otherwise unaffected nodes. It may also shift the residual baselines of other channels while leaving the local source of mismatch unresolved. This explains why common-mode correction performs well for the global offset, but poorly for the battery-local mismatch and structured ageing cases.

The main result of the mismatch analysis is that predictor validity is a dominant deployment condition for residual-based thermal AD. The fixed residual detector is effective when the expected-temperature predictor remains aligned, but small sustained biases can dominate the residual stream and drive cumulative evidence into long false-alert states. Common-mode correction suppresses purely global offset, but does not protect against node-dependent ageing or battery-local mismatch. Node-wise and hybrid correction keep false-alert occupancy near the baseline level in all tested mismatch scenarios while preserving complete alert-worthy recall.

Adaptive correction can expand the useful operating envelope by reducing mismatch-driven false-alert occupancy. However, adaptation introduces a different deployment risk: local correction can absorb local fault-like offsets if it is too free to update. It should therefore be treated as a bounded predictor-validity layer, not as an unrestricted anomaly-suppression mechanism. The correction state should also remain visible to operators because it provides evidence about predictor mismatch.

The operational conclusion is that residual-based on-board AD requires explicit predictor-validity management. A flight implementation would require update gating, anomaly-state freezing, local-bias diagnostics, operator-visible adaptive state fields, and mode-aware predictor management. This deployment interpretation is carried forward into the operational synthesis in Section 12.10.

12.9. Embedded Fixed and Adaptive Implementation Results

This section evaluates the STM32L4-class prototype described in Chapter 11. The firmware architecture, replay modes, packet classes, agreement workflow, and measurement sources are defined there; the purpose here is to report the measured embedded outcomes. The evaluation asks whether the implemented firmware preserves the operational event-lifecycle behaviour of the offline Python reference, remains within the timing, memory, and packet-size budgets, and demonstrates the node-wise adaptive-correction path under controlled predictor-mismatch replay.

The fixed embedded replay uses the on-board expected-temperature predictor before residual scoring. The adaptive embedded replay uses the same predictor followed by causal node-wise bias correction before residual scoring. The fixed replay covered all 243 event-centred evaluation windows. The adaptive replay covered 242/243 windows; the omitted window was benign, and all alert-worthy windows were included.

12.9.1. Replay Scope and Lifecycle Agreement

Offline-to-embedded agreement is assessed using the layered workflow defined in Section 11.8. Lifecycle agreement is treated as the primary implementation-parity metric because the operational output of the detector is a bounded event lifecycle. Exact equality of every terminal-event field is more fragile: event boundaries, causal spike suppression, nearby-gap flags, and audit metadata can differ slightly between the offline and embedded paths without changing the alerting behaviour that matters operationally.

Table 12.12: Offline-to-embedded lifecycle agreement after causal embedded spike suppression. Alert-worthy windows include telemetry-quality and thermal-fault events.

Window group	Windows	Lifecycle matches	Missing terminal events	Extra terminal events
Benign	206	187	1	22
Telemetry quality	30	29	0	1
Thermal fault	6	6	0	0
Alert-worthy total	36	35	0	1
Overall	242	222	1	23

The adaptive embedded replay matched the offline lifecycle sequence in 222/242 compared windows. Agreement was strongest for alert-worthy windows, where 35/36 lifecycle sequences matched. No alert-worthy embedded terminal events were missing. The only alert-worthy lifecycle mismatch was an abrupt calibration-step case on Panel Xm, where the embedded detector fragmented one offline event into two `START`/`CLEAR` pairs. This is interpreted as over-fragmentation rather than a missed alert-worthy event. The remaining mismatches were concentrated in benign windows and mainly reflect extra short embedded terminal events rather than loss of alert-worthy coverage.

12.9.2. Timing, Memory, Packet Size, and Compute Duty

The timing, memory, packet-size, and duty-cycle measurements use the instrumentation boundaries defined in Section 11.9. The fixed and adaptive embedded replays both remain far below the 15 s telemetry cadence and the 60 s alert-frame target. The fixed replay covered all 243 event-centred windows and 68 752 samples. The adaptive replay covered 242 windows and 68 416 samples. The combined resource results are summarized in Table 12.13.

Table 12.13: Embedded feasibility summary for fixed and adaptive STM32L4 replay.

Metric	Fixed on-board	Adaptive on-board	Interpretation
Replay coverage	243 windows, 68 752 samples	242 windows, 68 416 samples	The adaptive run omitted one benign window; all alert-worthy windows were included.
Mean processing time	0.586 ms/sample	0.600 ms/sample	Adaptive correction adds negligible absolute overhead relative to the 15 s telemetry cadence.
Worst observed sample	12.777 ms	12.797 ms	The worst observed adaptive path uses approximately 0.0853 % of the 15 s sample interval.
Routine update paths	Sub-millisecond	Sub-millisecond	Nominal, candidate-alarm, event-active, and gap/reset contexts remain below approximately 0.624 ms; the slow tail is caused by event-packet emission.
Memory footprint	55 880 B flash; 3328 B RAM	57 464 B flash; 3424 B RAM	Adaptive correction adds 1584 B flash and 96 B RAM, which does not change the memory-feasibility conclusion.
Packet sizes	Diagnostic: mean 82.6 B, max 96 B. Event: mean 114.4 B, max 132 B.		Operator-relevant event packets remain compact; diagnostic packets are replay-support artifacts.
Compute-duty estimate	Mean duty cycle: 0.003 91 %; worst path: 0.0852 %. Mean compute power: approx. 0.0014 mW.		Compute demand is negligible at prototype level, but this remains a timing-derived estimate rather than a direct flight-board power measurement.

Detailed timing, fixed-versus-adaptive runtime, runtime-context, memory, packet-size, and compute-duty measurements are provided in Section D.2 and Section D.3.

The main embedded result is that compute time, memory use, packet size, and estimated compute duty are not limiting factors for the prototype. Routine update paths remain sub-millisecond, while the worst observed timing tail is associated with event-packet emission. Even that path remains below 0.1 % of the 15 s sample interval. The remaining embedded risks are therefore not raw MCU feasibility, but integrated flight-software timing, sensor-bus interaction, storage and downlink integration, flight-board power measurement, and predictor-validity management.

The embedded results can also be placed in context relative to reported embedded AD implementations. The embedded implementation literature introduced in Section 2.4 is used here only as contextual evidence. The studies differ in platform, input cadence, sensor modality, algorithm family, validation setting, implementation scope, and output semantics. The comparison in Table 12.14 is therefore not a ranked benchmark. Its purpose is to situate the resource scale and output scope of the STM32L4 prototype relative to selected reported embedded AD implementations. The comparison is kept at summary level because the reported metrics are not directly benchmark-equivalent across studies.

Table 12.14: Contextual comparison with selected reported embedded AD implementations. Values are used to situate resource scale and output scope, not to rank performance. n.r. = not reported.

Study	Platform	Reported resource scale	Comparison role
This thesis	STM32L476	55.88 kB to 57.46 kB flash, 3.33 kB to 3.42 kB RAM, mean processing time 0.586 ms to 0.600 ms/sample, and worst observed path approximately 12.8 ms. Power is estimated only from timing duty cycle.	Reference implementation: bounded residual events, lifecycle handling, gap semantics, and compact alert packets.
Maununen [45]	MSP432	230.5 kB flash and 59.1 kB RAM reported, with possible flight reduction to 65.2 kB flash and 29.5 kB RAM. Inference time approximately 0.119 s. Power n.r.	Closest prior thesis context for embedded spacecraft temperature AD, but output is neural-network anomaly probability rather than detector-faithful event packets.
Horne et al. [29]	STM32H7	297.67 kB full-firmware flash, 53.83 kB SRAM, and mean execution time approximately 3.0 ms to 22.0 ms across clock settings. Full-testbed mean power approximately 44 mW to 99 mW.	Closest published spacecraft thermal-telemetry embedded comparator with reported timing, memory, and power context.
Moallemi et al. [46]	STM32L476	For the PCA configuration with compression factor 16: 32.82 kB to 91.04 kB flash, 11.12 kB to 77.55 kB RAM, 0.754 ms to 6.428 ms inference time, and 3.35 µJ to 73.96 µJ inference energy across input lengths.	Strong same-family STM32L4 resource comparator, but for structural-health vibration monitoring rather than spacecraft thermal telemetry.
Rosero-Montalvo et al. [56]	ARM IoT MCU	ARM implementation reports approximately 50 kB to 70 kB flash and approximately 12 kB to 13 kB RAM. Timing and energy n.r.	Useful compact ARM microcontroller memory comparator, but not spacecraft-specific and missing timing or power measurements.
Antonini et al. [1]	ESP32	Model file size approximately 56.9 kB to 283 kB; inference memory approximately 70 kB to 85 kB; inference time approximately 5 ms to 20 ms. Power n.r.	Useful embedded AD comparator with local inference and training, but platform, runtime, and monitoring task differ substantially.

The comparison supports a bounded interpretation. The proposed STM32L4 implementation lies within the resource scale of reported embedded AD systems and falls at the lower end of the reported RAM range in this contextual set. Relative to Maununen [45], the present work shifts the embedded output from neural-network anomaly-probability reporting toward detector-faithful residual evidence, event lifecycle handling, explicit gap semantics, and compact alert-packet generation. Power and energy values are not directly comparable across the cited studies because they are reported with different scopes, including full-testbed power, per-inference energy, and timing-derived duty-cycle estimates. The table should therefore be interpreted as contextual feasibility evidence rather than proof of superiority over prior embedded AD implementations.

Thoemel et al. [68] provide complementary spacecraft thermal-anomaly context through an in-orbit infrared-image payload, but are not included as a direct resource comparator because the implementation is Raspberry-Pi-based, image-based, and does not report MCU-style flash, RAM, inference latency, or energy-per-inference metrics.

12.9.3. Adaptive Correction on Hardware

The offline predictor-mismatch analysis in Section 12.8 evaluates the algorithmic value of adaptive correction. The embedded implementation chapter defines how predictor-mismatch replay is performed on the STM32L4 prototype. This subsection has a narrower purpose: to confirm that the node-wise adaptive predictor path is present in the firmware, updates causally during replay, is observable through diagnostic output, and behaves as intended under controlled local mismatch.

The targeted checks are therefore interpreted as functional embedded confirmation rather than flight validation. They test three properties that cannot be established from the offline analysis alone: update gating during fault-like residual evidence, convergence to a persistent local residual bias, and finite-bandwidth tracking of slow local drift.

Table 12.15: Embedded confirmation of node-wise adaptive correction under targeted STM32 replay checks. These checks confirm firmware behaviour under controlled replay; they do not establish a flight-ready adaptive correction policy.

Property checked	Embedded evidence	Interpretation
Adaptive state is observable and gated	Verbose replay exposed base prediction, adaptive bias, corrected prediction, residual, and update-gate state. During large fault-like residuals, updates were suppressed and bias remained small.	The adaptive path runs in firmware, and update gating prevents immediate learning of large fault-like residuals.
Persistent local offset can be re-centred	For a Batt +0.5 °C offset, fixed replay produced 8 detector events and 15 event packets; adaptive replay produced 1 detector event and 2 event packets. Final bias was 0.4958 °C, with residual median -0.0112 °C.	The node-wise adaptive path converges to a persistent local residual bias on hardware.
Slow local drift can be tracked	For a Batt ageing ramp from $0..1.3$°C over 83.33 h, fixed replay produced 56 detector events and 111 event packets; adaptive replay produced none. Final bias was 1.280 °C, with tracking MAE of 0.0123 °C.	The adaptive path can follow slow local predictor mismatch and reduce mismatch-driven alert burden.
Correction bandwidth is finite	Accelerated ramp replays produced increasing tracking lag as drift rate increased.	The adaptive layer does not instantly absorb rapid residual changes, helping preserve fast fault-like evidence.

The constant-offset and ageing-ramp checks show that adaptive correction is not merely compiled into the firmware; it changes the embedded detector behaviour in the expected direction. Under fixed prediction, the imposed Batt mismatch remains as persistent residual evidence and produces repeated detector events. Under node-wise adaptive prediction, the correction state converges toward the imposed offset or slow drift and reduces the resulting event burden.

The accelerated ramp result, shown in Figure 12.4, clarifies the correction bandwidth. Slow and moderate mismatch ramps are tracked closely, while faster compressed ramps produce increasing lag. This lag is a limitation if real predictor mismatch changes quickly, but it is also a safety-relevant property: the adaptive layer should compensate slow model drift or calibration mismatch, not immediately absorb fast residual changes that may represent genuine fault evidence.

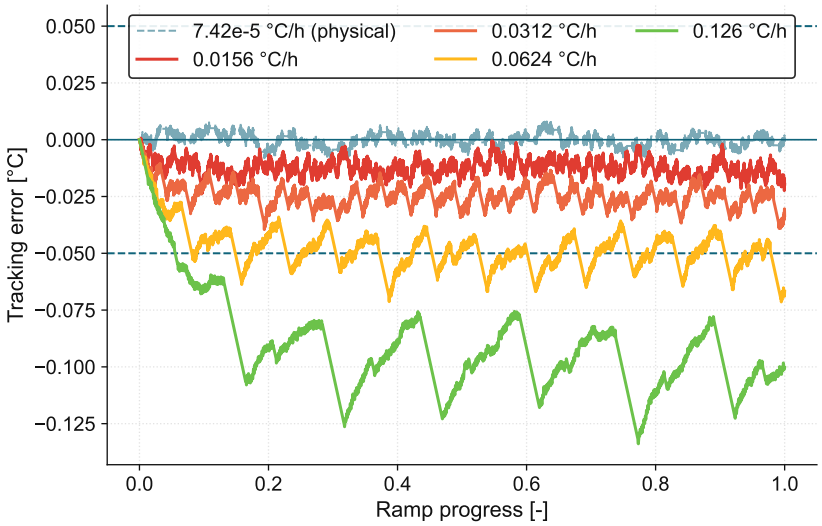


Figure 12.4: Embedded node-wise adaptive tracking error under accelerated Batt ageing-ramp mismatch.

These checks support a bounded embedded conclusion. Node-wise adaptive correction is implemented, observable, causal, and functionally active on the STM32L4 prototype. It can recentre persistent local residual bias and track slow local drift during controlled replay, while update gating suppresses adaptation during large fault-like residual evidence. However, the checks do not establish flight-safe adaptation. A flight implementation would still require broader validation across modes, nodes, telemetry-validity states, and fault families, together with operator-visible correction state and explicit freeze or fallback behaviour during anomaly evidence.

Detailed constant-offset and accelerated ageing-ramp tracking results are provided in Section D.4.

12.10. Operational Interpretation of the Evaluation Findings

The preceding sections evaluate the detector from several complementary perspectives: controlled labelled detection, decision-logic ablation, baseline comparison, predictor-mismatch robustness, and embedded fixed/adaptive implementation. Table 12.16 translates these technical findings into operational implications for an on-board advisory AD system. The table should not be read as a flight-readiness claim; it summarizes what the prototype evidence implies for operator awareness, alert burden, predictor validity, and embedded feasibility.

The evaluation also clarifies the relationship between the proposed pipeline and conventional threshold-based FDIR. The thesis does not argue that hard temperature limits should be removed. Upper/lower temperature limits remain necessary as deterministic safety and protection mechanisms. The contribution is instead an advisory residual-to-alert layer that can detect contextual deviations before, or without, crossing an absolute operating limit.

The ablation and F1 baseline results show why this advisory layer is not reducible to simple thresholding. Adaptive limits on observed temperature alone do not provide an acceptable trade-off between alert-worthy event recovery and alert burden. Residual threshold crossings alone are also insufficient as operational alerts: they require cumulative evidence, persistence, nuisance handling, and gap-aware lifecycle semantics before they become bounded detector-faithful alert events.

Overall, the results indicate that the residual-to-event pathway is suitable for continued engineering development and hardware-in-the-loop testing. The limiting factor is not STM32 compute capacity, but predictor validity and the operational governance of adaptive correction. These findings motivate the formal requirements verification and readiness assessment that follow.

12.11. Requirements Verification

The requirements verification matrix, Table 12.17, summarizes the demonstrated maturity of the prototype. *Pass* indicates that the requirement was satisfied within the STM32L4 replay prototype scope. *Partial* indicates that the function was demonstrated in replay but requires additional flight-context validation. *Not yet* indicates that the thesis identifies a required future capability rather than claiming deployment readiness.

The verification outcome supports a clear prototype-level conclusion. The implemented pipeline satisfies the core embedded feasibility objectives tested in this thesis: causal per-sample processing, compatibility with the 15 s telemetry cadence, large compute margin relative to a 60 s alert frame, compact detector-faithful packet output, and feasible prototype memory use. Under matched-predictor conditions, it also recovers all alert-worthy evaluation events.

The remaining work is flight integration and operational qualification rather than basic MCU feasibility. Predictor mismatch remains the main validity-management issue for the fixed detector, while adaptive correction is best interpreted as a proof-of-concept mitigation that requires further validation before flight use. A deployment path would require integration with the spacecraft FSW, on-board context reconstruction, parameter storage and version control, configuration update procedures, mode-aware validation, flight-board power measurement, and hardware-in-the-loop testing with flight-like telemetry flows.

Table 12.16: Operational interpretation of the main evaluation findings.

Question	Evaluation finding	Operational interpretation
Were alert-worthy synthetic events recovered?	The final detector recovered all 36/36 alert-worthy telemetry-quality and thermal-fault truth events under matched-predictor evaluation.	The residual-to-event pathway provides complete event coverage under controlled labelled conditions, but this should not be interpreted as flight recall.
How does the pipeline relate to conventional limit-based FDIR?	Temperature limits remain necessary for hard protection and Delfi-PQ heritage FDIR, but only detect predefined limit exceedance.	The proposed detector adds advisory context-aware monitoring by detecting temperatures abnormal relative to expected thermal behaviour, even without absolute limit violation.
Is simple thresholding sufficient?	The threshold-only baseline had lower recall, more detector events, more short events, and more alert packets than the final detector.	Bounded operational alerts require event lifecycle logic; raw threshold crossings are not sufficient as an on-board alert product.
Is a simple rules/statistics detector sufficient?	The causal F1 adaptive-limits baseline was either quiet but insensitive, or sensitive but operationally noisy.	Explicit thermal prediction, residual scoring, and event formation are needed for this deployment objective.
How often would operators see false alerts under matched assumptions?	The final detector produced 13 false detector events over 730.0 d, equivalent to about 0.018 false alerts/day.	Matched-predictor alert burden is low, but this result depends on predictor validity.
What is the main deployment risk?	Small sustained predictor mismatch can dominate the alert stream even when alert-worthy recall remains high.	Predictor validity, rather than embedded compute, is the main operational limitation of fixed residual detection.
Does adaptive correction remain embedded-feasible?	Node-wise adaptive prediction added approximately 0.013 ms/sample, 1.58 kB flash, and 96 B RAM.	Adaptive correction fits the STM32 prototype resource budget, but this does not make unrestricted adaptation flight-safe.
Does adaptive correction work on hardware under controlled mismatch?	In targeted Batt mismatch replays, the adaptive predictor corrected a $+0.5\,^{\circ}$C offset and tracked gradual accelerated ageing-ramp mismatch.	The adaptive path is active on hardware and can correct slow local model mismatch, but broader flight-like validation remains required.
Is MCU execution a blocker?	Routine update paths are sub-millisecond; worst observed samples were approximately 12.6 ms to 12.8 ms.	The compute path is far below the 15 s sample cadence and 60 s alert-frame budget; packet emission, not prediction, dominates the slow path.

12.12. Chapter Summary

This chapter evaluated the complete residual-to-alert prototype, from residual generation to embedded event reporting. Under matched-predictor conditions, the detector recovered all alert-worthy telemetry-quality and thermal-fault events in the evaluation split. It produced bounded alerts with a low false detector-event rate and a small alert-time fraction, indicating that the event lifecycle can provide sparse alerts for ground review rather than continuous alarm states.

The nuisance and gap results show that the detector does not treat all telemetry imperfections as alert-worthy events, and does not silently carry event state across telemetry gaps. Benign nuisance triggers remain, especially around isolated spikes and ordinary noise, but the resulting false-alert burden is low under matched prediction. Gap termination provides a conservative response to loss of observability by closing active events without claiming normal recovery.

Table 12.17: Requirements verification matrix for the embedded prototype.

Requirement / objective	Evidence	Status	Interpretation
Process 15 s telemetry samples	Event-window replay with causal per-sample processing	Pass	Demonstrated in prototype replay; not integrated with flight telemetry ingestion.
Fit within telemetry-frame compute margin	Worst observed processing time of 12.78 ms per sample	Pass	Compute path only; not a formal integrated FSW schedulability proof.
Recover alert-worthy events under matched predictor	36/36 alert-worthy truth events detected	Pass	Conditional on the expected-temperature predictor remaining aligned with telemetry.
Limit non-alert-worthy event burden	13 non-alert-worthy detector events over 730.0 d	Pass	Benign nuisance responses remain present but bounded in the synthetic evaluation.
Produce detector-faithful event packets	Event packet includes lifecycle phase, direction, residual magnitude, evidence flags, and clearance reason	Pass	Packet fields support ground review; operator study not performed.
Handle telemetry gaps causally	Gap reset and `GAP_TERMINATED` lifecycle behaviour	Pass	Demonstrated in replay windows; broader dropout patterns require further testing.
Fit within prototype memory budget	55.1 kB flash and 3.3 kB RAM	Pass	Prototype firmware footprint, not a complete integrated flight-software build.
Maintain low compute duty cycle	Processing time relative to the 15 s telemetry cadence	Partial	Timing-derived estimate only; direct current or energy measurement was not performed.
Remain robust under predictor mismatch	Predictor-mismatch stress tests with fixed and adaptive predictors	Partial	Fixed predictor is mismatch-limited; adaptive correction reduces selected mismatch-driven alert burden but is not flight-qualified.
Integrated flight readiness	STM32L4 prototype with UART replay workflow	Not established	Requires FSW integration, flight-interface testing, power measurement, and flight-environment validation.

The predictor-mismatch results identify predictor validity as the main deployment condition for residual-based thermal AD. The fixed-predictor detector is operationally trustworthy only while the expected-temperature model remains aligned with observed telemetry. Small sustained biases can dominate cumulative evidence and produce persistent false-alert states. Adaptive correction can substantially reduce this burden in validation stress tests, but local correction can also absorb fault-like offsets. Adaptation must therefore be bounded, gated, and visible to operators before flight use.

The embedded results show that MCU feasibility is not the limiting factor under the tested prototype conditions. The STM32L4 implementation runs far below the 15 s sample cadence and 60 s alert-frame budget, with small flash and RAM use and compact event packets. The prototype is therefore suitable for continued hardware-in-the-loop and mission-integration testing. It does not yet establish beacon-level operational trust without additional predictor-validity monitoring, mode-aware calibration, and flight-like validation.

13

Real-Telemetry Stress Tests on FUNcube-1 Telemetry

The labelled synthetic benchmark in Chapter 12 provides controlled event-level evaluation because anomaly onset, offset, affected node, subtype, and alert-worthiness are known by construction. FUNcube-1 telemetry provides a complementary evidence layer: it exposes the workflow to real on-orbit temperature behaviour, including incomplete observations, contact gaps, telemetry artifacts, partial-orbit coverage, contextual thermal regimes, and orbit-scale thermal texture.

This chapter uses selected FUNcube-1 temperature-telemetry windows as qualitative real-telemetry stress tests. The purpose is to assess whether the prediction, residual-scoring, detector-evidence, and event-state inspection workflow remains interpretable on real spacecraft telemetry, and whether behaviours represented in the synthetic benchmark have plausible real-telemetry counterparts. Detector intervals are therefore interpreted as detector-declared anomalous residual events, not as confirmed spacecraft faults. Accordingly, this chapter does not report real-data recall, precision, false-alert rate, or Time to Awareness (TTA).

The chapter first describes the context reconstruction required to make selected FUNcube windows scoreable. It then presents four stress-test categories: telemetry-quality artifacts, contextual thermal-regime deviations, gap-limited partial observability, and apparent ripple-like regime structure. The chapter closes by comparing the FUNcube evidence with the synthetic benchmark and identifying what the real-telemetry cases add to the deployment-readiness assessment.

13.1. Raw-Timeline Thermal Context Reconstruction

The FUNcube stress tests require thermal context on the same timeline as the selected raw telemetry windows. Several target intervals contain usable temperature telemetry, but the corresponding prefiltered context products do not always provide valid thermal phase information at those exact samples. Discarding these windows would remove useful real-telemetry stress cases.

To preserve them, thermal context was reconstructed directly on the raw telemetry timeline. Empirical photo-current eclipse detection was used to identify eclipse and sunlight transitions and define a sunrise-to-sunrise thermal phase. Skyfield/TLE propagation supplied orbital geometry and solar-distance context. This reconstruction makes the selected windows scoreable by the residual-inspection workflow, while keeping the FUNcube analysis qualitative rather than label-based. The resulting context fields are summarized in Table 13.1.

The reconstructed context assigns finite thermal phase to more than 2.5×10^6 raw telemetry rows. The median empirical orbit period of 98.0 min is consistent with LEO thermal cycling, while the median transition angle of approximately 257° reflects the adopted phase convention: $\theta = 0^\circ$ marks eclipse exit and the transition near $\theta \approx 257^\circ$ marks the onset of cooling or eclipse. These fields provide empirical thermal context for scoring selected windows, not exact spacecraft attitude knowledge.

Table 13.1: Raw-timeline FUNcube thermal context reconstructed for qualitative real-telemetry stress testing.

Quantity	Value
Raw telemetry rows	4 217 924
TLE records	6087
Time span	2013-11-20 to 2023-01-12
Rows with finite thermal phase	2 562 532
Raw eclipse fraction	0.1840
Clean eclipse fraction	0.1837
Cooling-branch fraction	0.1633
Median thermal transition angle	257.14°
Median empirical orbit period	98.0 min

13.2. FUNcube Processing Workflow

Each FUNcube case was processed using the workflow summarized in Table 13.2. The workflow starts from raw temperature telemetry, reconstructs thermal phase and orbital context, fits a local expected-temperature reference, computes residuals and detector evidence, and inspects the resulting event-state behaviour. This mirrors the thesis residual-to-alert workflow at the level needed for real-telemetry inspection, while keeping the interpretation qualitative because independent fault labels are unavailable.

Where appropriate, a constrained sinusoidal ripple term was used as a nuisance model to improve residual interpretability. This term is not treated as an attitude or spin-state model. Its purpose is only to separate repeatable ripple-like thermal texture from larger contextual residual deviations. Contact gaps are represented explicitly using context-only gap rows, so the detector can reset evidence during loss of observability rather than treating timestamp jumps as continuous telemetry.

Table 13.2: Processing workflow used for the FUNcube real-telemetry stress tests. The workflow makes residual and detector-evidence traces inspectable on real telemetry; it is not used to produce labelled detection metrics.

Stage	Purpose	Output used for interpretation
Raw telemetry timeline	Use raw FUNcube observations as the master timeline so that partially observed target intervals are preserved.	Observed engineering-unit temperature samples and real contact gaps.
Thermal context reconstruction	Assign empirical sunrise-to-sunrise thermal phase using photo-current eclipse timing, supplemented by orbital geometry where needed.	Thermal phase, heating/cooling branch context, solar-distance context, and orbital context.
Local expected-temperature fitting	Fit an expected-temperature model using surrounding data outside the target interval and guard buffer.	Predicted temperature for the target window.
Constrained ripple nuisance modelling	Optionally model repeatable panel-ripple structure without claiming knowledge of true attitude or spin state.	Ripple-adjusted prediction and residuals where applicable.
Gap-row insertion	Represent contact gaps explicitly as loss of observation rather than continuous telemetry.	Context-only gap rows with no observed temperature, residual, or normalized residual.
Residual detection and inspection	Run the residual detector and inspect residuals, detector evidence, event state, contact gaps, and declared intervals.	Detector-declared anomalous residual events for qualitative interpretation.

13.3. Case-Study Selection

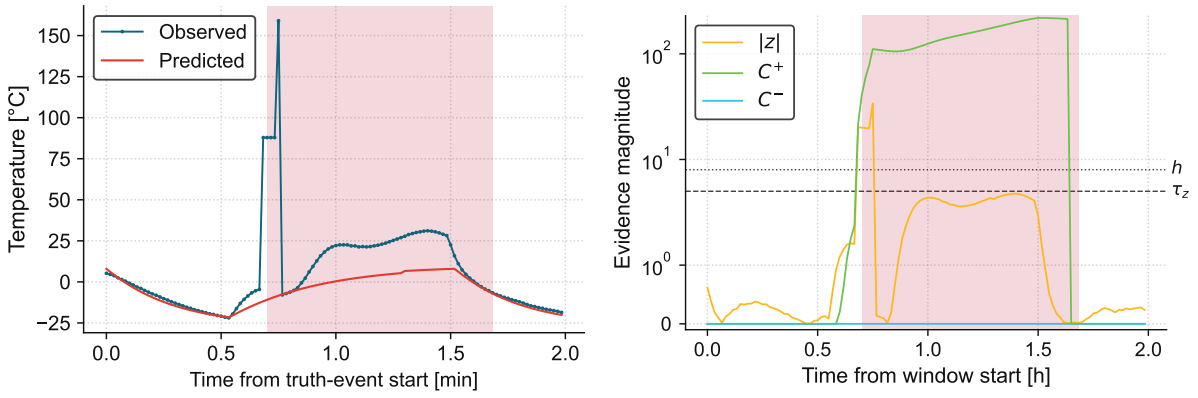
The selected FUNcube windows are chosen to stress different parts of the residual-detection workflow. They are not labelled as true or false spacecraft faults. Instead, each window is assigned a qualitative stress-test role, summarized in Table 13.3.

Table 13.3: FUNcube-1 real-telemetry stress-test windows and their intended interpretation. Detector outputs in these windows are interpreted as detector-declared anomalous residual events, not as independently confirmed faults.

Window	Channel	Real-telemetry stressor	Main lesson
2014-09-04 pinned / spike / saturation	Solar Panel $-Y$	Telemetry-quality artifact with a large spike and abnormal segment	Artifacts remain visible in the residual stream and are not fitted away.
2019-05-12 contextual hot/cool orbits	Silver panel	Hotter-than-expected contextual thermal behaviour	Real thermal context can dominate residual evidence without independently confirming a hardware fault.
2015-09-14 gap-limited cooler orbit	Solar Panel $-X$	Cooler-than-expected partial orbit with contact gaps	Partial observability affects residual interpretation and detector-state evolution.
2014-08-07 fast ripple	Solar Panel $-X$	Fast panel ripple reference regime	Spin- or illumination-imprinted ripple can appear as coherent nuisance residual structure.
2014-08-21 intermediate ripple	Solar Panel $-X$	Intermediate apparent spin/tumble regime	Less well-modelled ripple can produce repeated residual excursions and detector evidence.
2014-09-06 slow ripple	Solar Panel $-X$	Slow ripple and orbit-scale regime mismatch	Attitude or illumination-regime changes can dominate residual evidence.

13.4. Telemetry-Quality Artifact Case

The first FUNcube stress test uses a Solar Panel $-Y$ window from 2014-09-04 containing an abnormal high-temperature segment and spike. This case supports the telemetry-realism basis for including isolated spikes, saturation-like behaviour, and telemetry-quality artifacts in the synthetic benchmark. The purpose is to test whether abnormal measurement behaviour remains visible in the residual stream rather than being absorbed by the expected-temperature fit.



(a) Observed and predicted temperature.

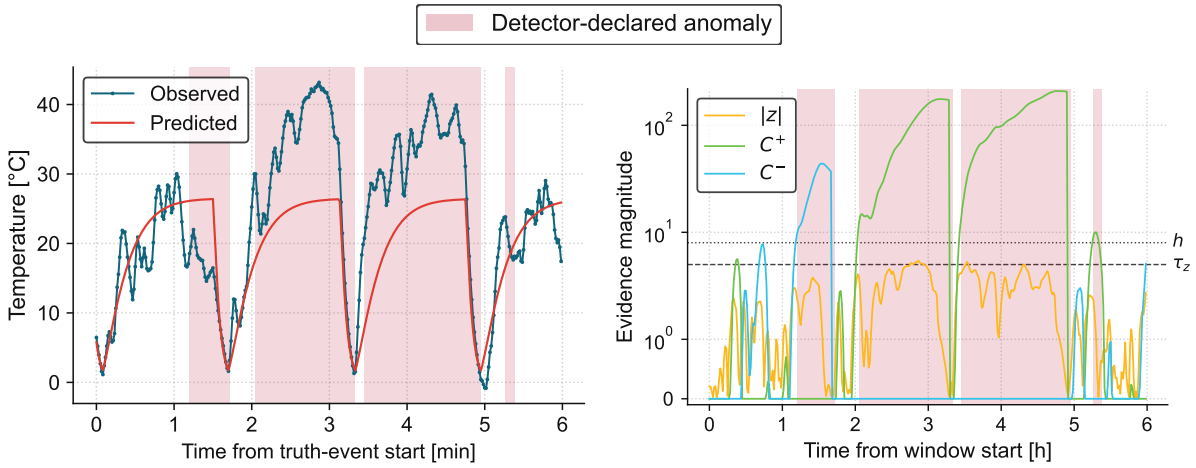
(b) Detector evidence.

Figure 13.1: FUNcube real-telemetry stress test showing a telemetry-quality artifact in the Solar Panel $-Y$ temperature channel on 4 September 2014.

Figure 13.1 shows that the abnormal segment remains visible in the observed temperature, residual, and detector-evidence traces. This supports the benchmark assumption that telemetry-quality artifacts can produce detector-relevant residual structure and should be represented separately from physical thermal faults. The declared interval is interpreted as a detector-declared residual event, not as an independently confirmed FUNcube fault.

13.5. Contextual Thermal-Regime Case

The second FUNcube stress test uses a silver-panel window from 2019-05-12 in which the observed temperature becomes substantially warmer than the fitted expected-temperature waveform over several orbits. This case supports the benchmark assumption that real spacecraft temperature telemetry can contain contextual thermal-regime changes that dominate residual evidence even when the underlying cause is not labelled.



(a) Observed and predicted temperature.

(b) Detector evidence.

Figure 13.2: FUNcube real-telemetry stress test showing contextual hot-orbit behaviour in the silver-panel temperature channel on 12 May 2019.

Figure 13.2 shows that the warmer-than-expected interval appears as sustained positive residual evidence and produces detector-declared residual events. This supports the need to include contextual thermal-regime stressors and predictor-validity concerns in the evaluation framework. The available FUNcube data do not establish whether the cause is an attitude or illumination change, an operational regime, a measurement issue, or a physical fault, so the result is interpreted as workflow-level evidence rather than fault validation.

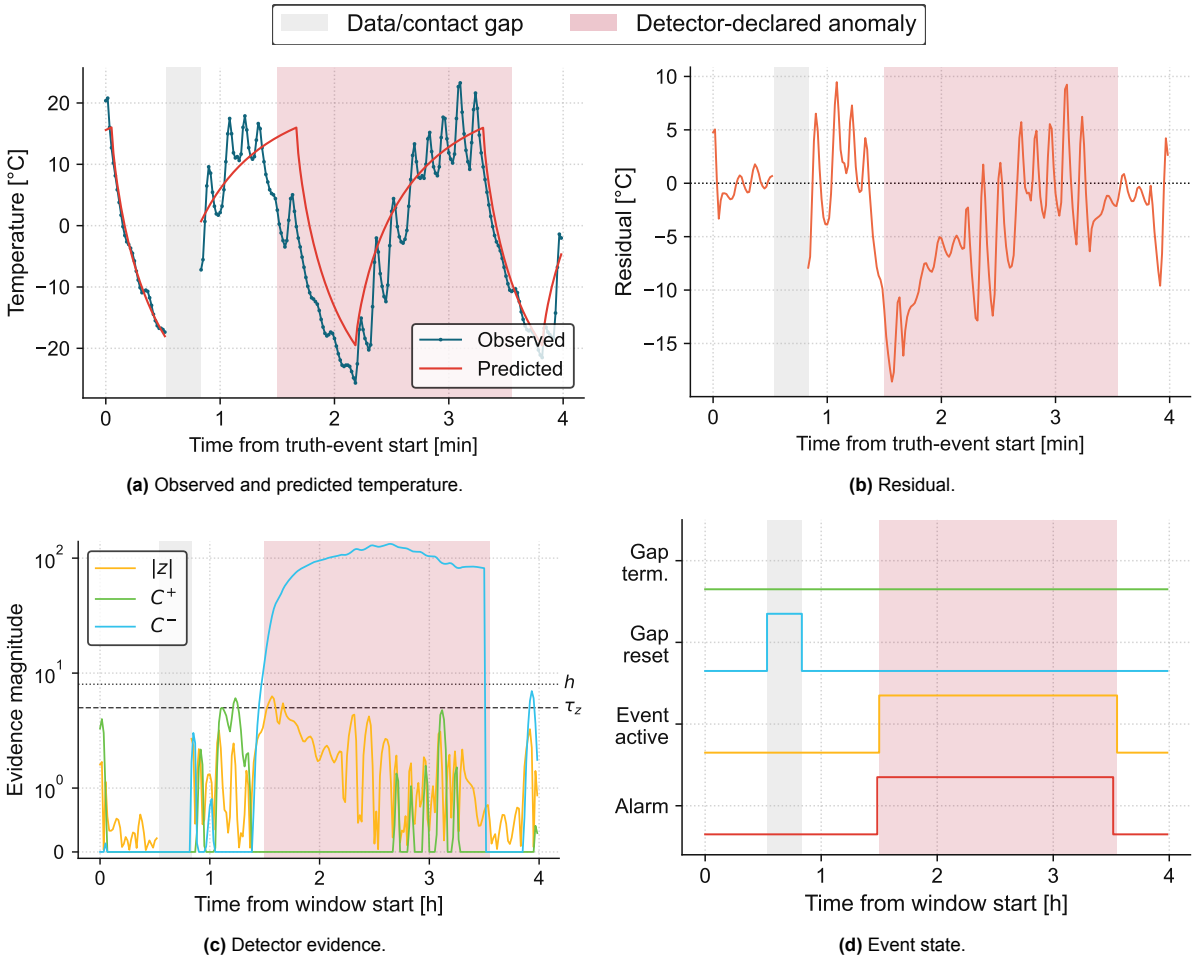
13.6. Gap-Limited Partial-Observability Case

The third FUNcube stress test uses a Solar Panel $-X$ window from 2015-09-14 containing a cooler-than-expected partial orbit and a contact gap. This case supports the benchmark assumption that gaps and partial observability are first-order telemetry conditions, not minor formatting issues. It tests whether prediction context can remain available while residual evidence is correctly removed where observations are absent. For this case, 18 context-only gap rows were inserted at a 60 s cadence across the contact gap, using a 180 s gap threshold. This preserves a finite prediction context while marking the observed temperature, residual, and normalized residual as unavailable during loss of observability.

Figure 13.3 shows that the cooler-than-expected partial orbit remains visible as negative residual evidence, while the contact gap is treated as loss of observability. This supports the synthetic benchmark choice to represent gaps explicitly and to handle them separately from physical recovery or continued fault evidence. The case also shows why detector states must preserve observability information: a finite prediction through a gap is not the same as a valid residual.

13.7. Spin/Ripple Regime Cases

The final FUNcube stress tests compare three Solar Panel $-X$ windows from 2014-08-07, 2014-08-21, and 2014-09-06. These windows occur close together in time but show different apparent panel-ripple regimes. This case supports the benchmark assumption that panel temperatures can contain coherent ripple or illumination-imprinted thermal texture that affects residual evidence.



(a) Observed and predicted temperature.

(b) Residual.

(c) Detector evidence.

(d) Event state.

Figure 13.3: FUNcube real-telemetry stress test showing a gap-limited cooler-than-expected orbit in the Solar Panel $-X$ temperature channel on 14 September 2015.

The interpretation is deliberately limited. No independent attitude or spin-state truth is used here, so these windows are not labelled as spin anomalies or attitude faults. Their value is morphological: they show that real panel telemetry can contain ripple-like structure that becomes detector-relevant when it is not captured by the expected-temperature model.

Figure 13.4 shows that panel ripple and orbit-shape mismatch can appear as coherent residual structure and detector evidence. This supports the benchmark choice to include nominal texture and ripple-like nuisance structure, and it reinforces why residual alerts should be interpreted as departures from the expected model rather than root-cause claims. In operational use, apparent spin, tumble, attitude, or illumination-regime effects would require contextual review before being treated as physical thermal faults.

13.8. Relationship Between the Synthetic Benchmark and FUNcube Real Telemetry

The FUNcube stress tests provide morphology-level and workflow-level support for selected assumptions in the synthetic benchmark. They do not validate the benchmark as flight-accurate Delfi Twin telemetry, and they do not provide labelled detector-performance metrics. This distinction is important because the two evidence sources serve different purposes. The synthetic benchmark provides controlled event labels, including onset, offset, affected node, subtype, and alert-worthiness. FUNcube provides real on-orbit temperature telemetry with imperfect observability, contextual thermal variation, telemetry-quality artifacts, contact gaps, partial coverage, and panel ripple structure.

FUNcube can therefore support the *relevance* of several benchmark behaviours, but not their exact Delfi Twin magnitudes, frequencies, labels, or operational causes. It is most informative for external or panel-related thermal morphology. It is much less informative for internal fault families, such as battery over-temperature, MCU self-heating, heater-control failure, or internal conductance changes, because the selected FUNcube telemetry does not provide the same internal sensor coverage assumed in the Delfi Twin benchmark.

A second limitation concerns temporal resolution and observability. Some benign nuisance behaviours in the synthetic benchmark are deliberately low-amplitude or short-timescale effects. Small quantization and ordinary low-amplitude noise may not be visually or statistically separable in the available FUNcube warehouse telemetry, where they can be masked by orbit-scale thermal evolution, panel ripple, interpolation effects, and sparse temporal coverage. Their lack of clear visual confirmation in FUNcube should therefore not be interpreted as evidence against including them as controlled robustness cases.

Table 13.4 summarizes the relationship between the synthetic benchmark and the selected FUNcube stress-test evidence. A more detailed behaviour-by-behaviour mapping is provided in Appendix E.

Table 13.4: Relationship between synthetic benchmark assumptions and FUNcube real-telemetry stress-test evidence.

Evidence category	FUNcube support	Claim boundary
Orbit-scale thermal behaviour	FUNcube shows repeated heating and cooling waveforms suitable for testing local expected-temperature fitting.	Supports plausible nominal morphology only; does not prove flight-accurate Delfi Twin temperatures.
Panel ripple and illumination-imprinted texture	Selected panel windows show fast, intermediate, and slow ripple-like regimes that affect residual evidence.	Supports inclusion of residual texture and panel-level nuisance structure; no independent attitude or spin-state truth is available.
Contact gaps and partial observability	Gap-limited windows show that real telemetry contains incomplete observability and contact interruptions.	Supports explicit gap handling; does not provide labelled recall or precision across gaps.
Telemetry-quality artifacts	Selected windows contain abnormal telemetry excursions and spike-like behaviour visible in residual evidence.	Supports separate handling of telemetry-quality artifacts; does not establish their frequency, cause, or operational rate.
Internal thermal fault families	Battery, MCU, heater-control, and internal-conductance fault families are not directly observable in the selected FUNcube channels.	These remain mission-scoped synthetic fault cases requiring Delfi Twin-like internal sensing or independent labels.
Residual-to-event workflow	The full prediction, residual, evidence, gap, and event-inspection workflow can be exercised on messy real telemetry.	Qualitative workflow stress test only; not labelled detector validation.

Overall, FUNcube supports the synthetic benchmark at the level of selected telemetry morphologies and detector-workflow plausibility. It provides real examples of orbit-scale thermal structure, panel ripple, incomplete observability, contextual thermal variation, and telemetry-quality artifacts. It does not provide the internal sensor coverage or independent fault labels needed to validate Delfi Twin battery, MCU, heater, or internal self-heating fault classes. The synthetic benchmark therefore remains necessary for quantitative evaluation, while FUNcube provides a real-telemetry check on whether the benchmark stressors and residual-to-event workflow remain operationally plausible under non-ideal on-orbit conditions.

13.9. Synthesis

The FUNcube stress tests show that the residual-to-event workflow remains inspectable on real spacecraft telemetry. The selected windows produce interpretable temperature predictions, residuals, detector evidence, and event-state traces despite telemetry artifacts, contact gaps, partial-orbit coverage, contextual thermal variation, and apparent spin or illumination-imprinted panel ripple.

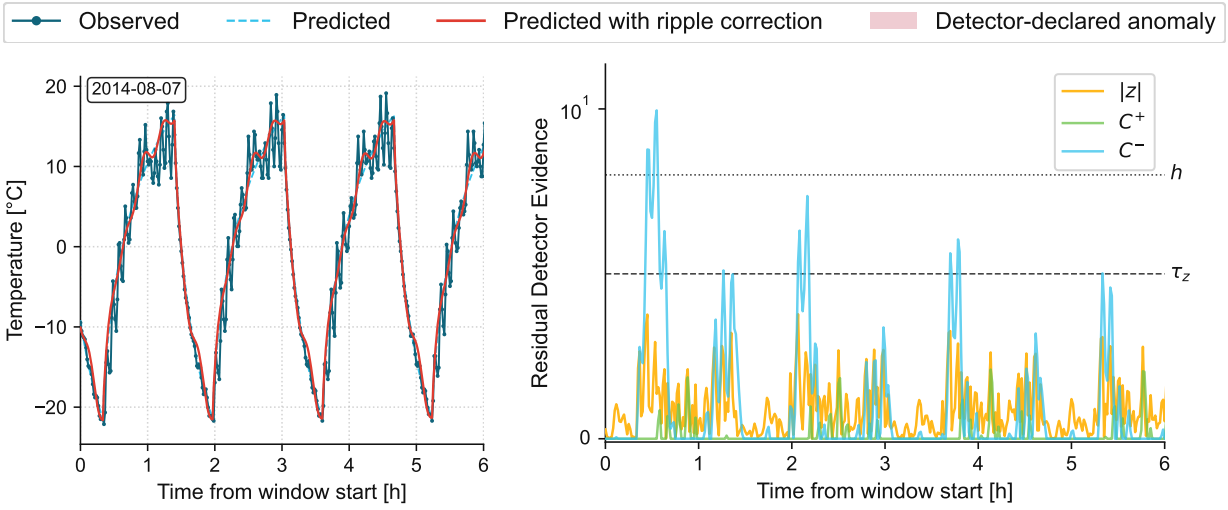
Table 13.5: Detector-declared FUNcube residual events from the selected real-telemetry stress-test windows.

Window	Channel	Declared events	Dominant residual behaviour	Interpretation
2014-09-04 telemetry-quality artifact	Solar Panel $-Y$	1	Large positive residual associated with an abnormal telemetry excursion and spike.	Telemetry-quality residual event; not interpreted as a confirmed thermal hardware fault.
2019-05-12 contextual hot orbits	Silver panel	8	Sustained positive residual during warmer-than-expected orbit-scale behaviour.	Contextual residual events requiring operator and context review.
2015-09-14 gap-limited cooler orbit	Solar Panel $-X$	3	Sustained negative residual around a partial orbit, interrupted by a contact gap.	Gap-limited residual events shaped by partial observability.
2014-08-07 fast ripple regime	Solar Panel $-X$	0	Coherent high-frequency panel ripple with comparatively bounded residual evidence.	Reference ripple/nuisance regime without detector-declared events.
2014-08-21 intermediate ripple regime	Solar Panel $-X$	7	Larger ripple or regime mismatch causing repeated residual excursions.	Detector-declared residual events consistent with changed apparent spin or illumination regime.
2014-09-06 slow ripple regime	Solar Panel $-X$	6	Stronger orbit-scale and ripple mismatch relative to the fitted expected structure.	Residual events consistent with apparent attitude or illumination-regime mismatch, not a confirmed fault.

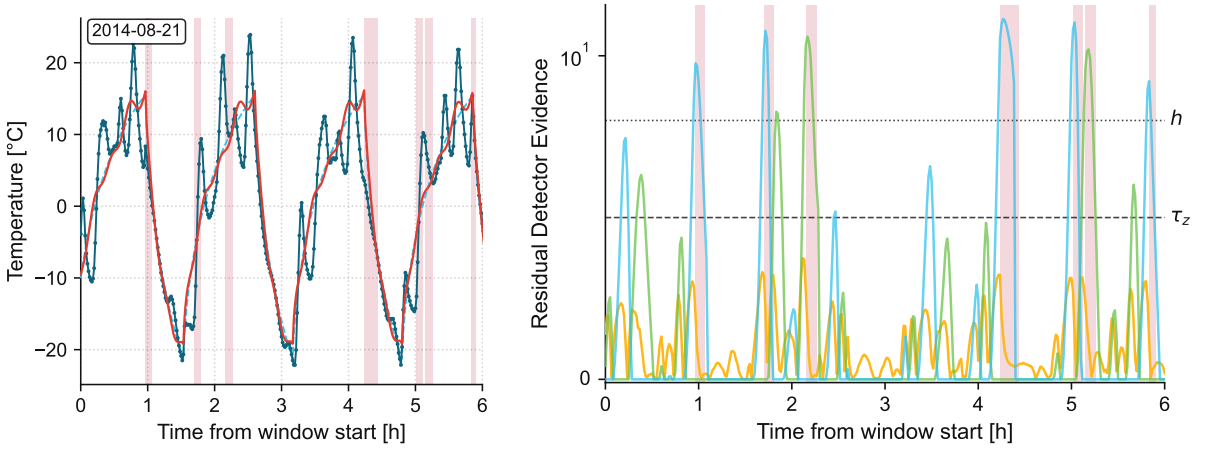
The detector-declared outputs from the selected stress-test windows are summarized in Table 13.5. The table shows that the workflow does not respond uniformly to all real-telemetry texture: the fast-ripple reference window produced no detector-declared events, while the contextual, gap-limited, telemetry-quality, and stronger ripple-regime windows produced bounded residual events.

These outputs are qualitative residual-event behaviour, not labelled true positives or false positives. The cases therefore complement the synthetic benchmark rather than replacing it. Synthetic evaluation provides labelled recall, false-alert, and time-to-awareness metrics under controlled conditions; FUNcube provides realism and exposes operational complications that must be handled before flight use.

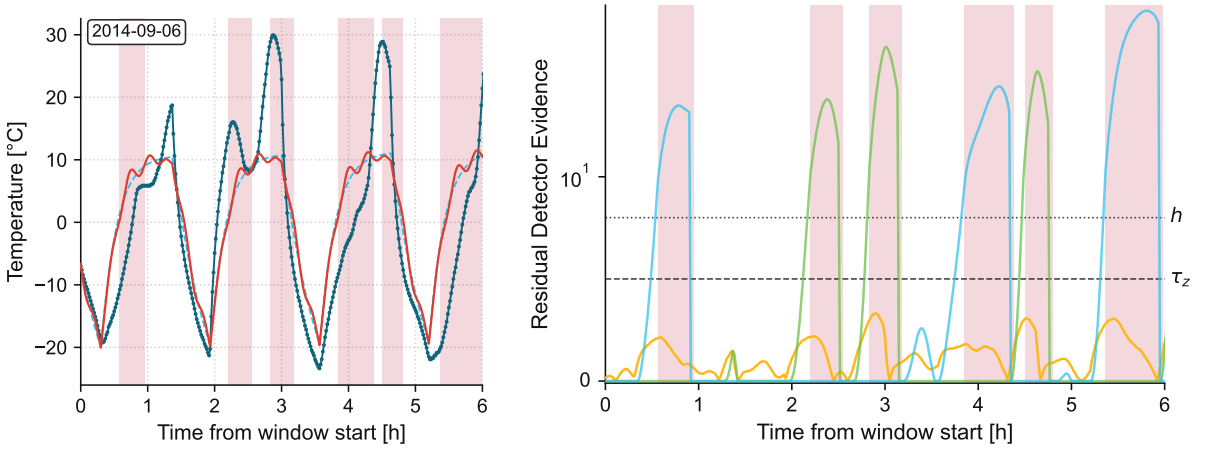
The main conclusion is that real telemetry makes the residual-event output interpretable but context-dependent. Operator triage remains necessary, especially when attitude, illumination, telemetry-quality, or sensor-validity context is incomplete.



(a) Fast ripple regime.



(b) Intermediate ripple regime.



(c) Slow ripple and stronger orbit-shape mismatch.

Figure 13.4: FUNcube real-telemetry stress tests showing fast, intermediate, and slow panel-ripple regimes in Solar Panel $-X$ temperature telemetry.

Part IV:

Significance, Deployment Value, and Conclusions

Building on the implementation and evaluation results of Part III, this part interprets the work from an operational perspective. It assesses mission usefulness, claim boundaries, limitations, and deployment readiness, and concludes by summarizing the thesis contributions and recommended next steps.

14

Operational Value, Limitations, and Deployment Readiness

The previous chapters evaluated the residual-to-event Anomaly Detection (AD) pipeline using controlled synthetic labels, decision-logic ablations, predictor-mismatch stress tests, embedded STM32L4 replay, and FUNcube-1 real-telemetry stress cases. This chapter interprets those results from a deployment perspective. It does not treat the prototype as a flight-ready system. Instead, it identifies the operational value supported by the evidence, the boundaries of the supported claims, and the additional work required before flight use.

The central conclusion is that the prototype is credible as a deployment-oriented advisory anomaly-awareness system, but not yet as a flight-trusted autonomous capability. The detector can produce compact, interpretable thermal event alerts under controlled assumptions and is feasible on STM32L4-class hardware. The limiting issue is not raw embedded compute capacity, but operational trust under predictor mismatch, mode changes, telemetry imperfections, adaptive-correction governance, and unvalidated operator workflows.

Deployment-Oriented Workflow Step

8. Assess operational significance, deployability, and future pathway
Interpret the mission usefulness and embedded feasibility, identify the main limitations and residual risks, and assess the prototype's readiness for continued development, hardware-in-the-loop validation, and eventual flight integration.

14.1. Operational Interpretation of the Results

The evaluation supports five operational conclusions.

First, the residual-to-event pathway works under matched-predictor assumptions. In the held-out synthetic evaluation, the final detector recovered all alert-worthy telemetry-quality and thermal-fault events while producing a bounded event stream. This supports the use of residual evidence, cumulative evidence, nuisance handling, and lifecycle logic when the expected-temperature model remains aligned with the telemetry.

Second, the selected detector is justified relative to the simpler alternatives tested here. The ablation and baseline studies show that the final detector is not reducible to simple residual thresholding or causal adaptive limits. Threshold-only detection produced more fragmentation and nuisance escalation, while the adaptive-limits baseline could not provide the same recall-alert-burden trade-off for the tested deployment objective.

Third, embedded compute is not the main prototype-level barrier. Fixed and adaptive STM32L4 replay remained far below the 15 s telemetry cadence, with low memory use and compact packet sizes. The embedded result therefore supports continued development of the detector core on MCU-class hardware.

Fourth, adaptive correction is promising but safety-sensitive. The adaptive path can reduce mismatch-driven alert burden and was confirmed on hardware under targeted local mismatch replay. However, adaptation cannot be treated as automatically safe, because local correction may absorb fault-like residual evidence if not bounded, gated, frozen during anomaly evidence, and visible to operators.

Fifth, predictor validity is the dominant deployment risk. A residual event is meaningful only while the expected-temperature model remains valid for the current operating context. If the predictor is stale, biased, or mode-mismatched, residual evidence may reflect predictor invalidity rather than abnormal spacecraft behaviour. A flight implementation therefore requires predictor-validity monitoring, mode-aware configuration, and operator-visible diagnostic state.

14.2. Mission Value and Supported Claims

The prototype's mission value lies in converting continuous thermal telemetry deviations into bounded, detector-faithful event alerts. If integrated with the flight telemetry and downlink workflow, this could support earlier operator awareness, selective downlink of event summaries and context windows, and transparent triage evidence without claiming on-board diagnosis or autonomous recovery.

Table 14.1 consolidates the main supported claims, their evidence basis, and their remaining boundaries. The table is intentionally claim-limited: it states what the thesis supports at prototype level, while separating those claims from unsupported flight-readiness or autonomous-trust claims.

The central claim boundary is therefore that this thesis supports prototype-level behaviour claims under controlled synthetic labels and STM32L4 replay; it does not validate flight performance, real-fault performance statistics, or autonomous operational trust.

These claims are cumulative rather than dependent on a single headline metric. The synthetic benchmark supports controlled event-level performance claims. The embedded replay supports Microcontroller Unit (MCU)-feasibility claims. The predictor-mismatch study identifies the main operational validity risk. The FUNcube-1 cases support real-telemetry interpretability, but not labelled validation. Together, these results justify continued development of the prototype, not direct flight deployment.

However, the supported claims remain conditional. In particular, residual-event outputs are meaningful only when the telemetry, predictor, adaptation logic, and event morphology remain within the tested prototype envelope. Table 14.2 makes this interpretation boundary explicit before the deployment-readiness assessment.

This envelope is the basis for the readiness judgement below. The prototype is useful when these assumptions are satisfied or monitored, but direct flight use would require safeguards for cases where they are violated.

14.3. Deployment Readiness Assessment

The prototype should be interpreted as an embedded proof of concept and deployment-readiness study, not as a flight-ready AD system. Several components are mature enough to carry forward, while others require additional integration and validation.

Table 14.1: Supported prototype-level claims, evidence basis, and remaining claim boundaries.

Supported prototype-level claim	Evidence basis in this thesis	Not claimed / remaining boundary
Advisory on-board anomaly awareness is an appropriate operational role.	Fault-management allocation, mission-use-case framing, and alert-payload design.	Does not replace diagnosis, response selection, adaptation management, or existing deterministic Fault Detection, Isolation and Recovery (FDIR).
Bounded residual-to-event alerts can support earlier thermal-event awareness.	Held-out synthetic event-level evaluation, lifecycle behaviour, and time-to-awareness results.	Supported under controlled labelled conditions; not a real-flight recall, precision, or false-alert-rate claim.
Compact event packets can support downlink triage.	Alert-payload definition, event packet implementation, and packet-size results.	Final downlink scheduling, storage policy, prioritization rules, and operator display remain future work.
Detector-faithful evidence can support operator triage.	Packet fields report affected node, timing, direction, residual magnitude, evidence flags, and clearance reason.	The packet explains detector evidence, not physical root cause.
The selected lifecycle is justified relative to tested simpler alternatives.	Decision-logic ablation and causal F1 adaptive-limits baseline comparison.	Does not reject all alternative detectors; only supports this design relative to the tested comparators.
The detector core is feasible on STM32L4-class hardware at prototype level.	Embedded replay, measured timing, memory, packet-size, and adaptive confirmation results.	Does not prove integrated flight-software timing, sensor-bus behaviour, persistent storage, downlink scheduling, or flight-board power.
Real telemetry can exercise the workflow qualitatively.	FUNcube-1 stress tests show inspectable prediction, residual, evidence, gap, and event-state behaviour on non-ideal real telemetry.	Qualitative real-telemetry stress evidence only; no labelled recall, precision, false-alert rate, or confirmed root-cause validation.

Table 14.2: Operational validity envelope for interpreting residual-event outputs. The detector output is meaningful only when the underlying telemetry, predictor, and adaptation assumptions remain inside the tested prototype envelope.

Condition	Inside the tested envelope	Outside the envelope
Predictor context	Expected-temperature model is aligned with the current thermal mode and operating context.	Flag predictor invalidity, disable scoring, switch model, or require recalibration before interpreting residual events.
Telemetry observability	Temperature samples are available, finite, within the configured engineering range, and causally ordered. Gaps are marked explicitly.	Use gap termination or reset semantics; do not infer event continuity through missing telemetry.
Adaptive correction	Correction is bounded, gated, slow, and visible to operators. Updates are suppressed during strong anomaly evidence.	Freeze or limit adaptation; do not allow local correction to silently absorb fault-like residual evidence.
Event morphology	Detected behaviour resembles tested nuisance, telemetry-quality, or thermal-fault families.	Treat outputs as advisory evidence only; do not generalize recall, false-alert rate, or root-cause interpretation beyond tested cases.

The recommendation is:

Deployment Readiness Recommendation

Supported for continued prototype development and hardware-in-the-loop validation.

Not yet supported for direct flight deployment or autonomous beacon-style trust.

The prototype demonstrates that the residual-to-event pathway is feasible and operationally useful under controlled assumptions. However, predictor validity, mode-aware configuration, adaptive-correction governance, integrated flight-software timing, flight-board power, and operator-facing validation remain unresolved.

Table 14.3 summarizes the readiness judgement by capability. The purpose is not to produce a binary pass/fail decision, but to separate what can be carried forward from what still requires validation.

Table 14.3: Deployment readiness summary for the prototype.

Capability	Readiness status	Required next step
Matched-predictor residual detection	Supported at prototype level	Validate under more realistic predictor mismatch, mode changes, and real or flight-like telemetry.
Event lifecycle and alert packets	Supported at prototype level	Integrate final storage, downlink prioritization, and operator display.
Embedded timing and memory	Supported at prototype level	Remeasure under integrated Flight Software (FSW), interrupt load, bus activity, storage, and downlink operations.
Telemetry-gap handling	Supported at prototype level	Connect to flight telemetry-validity flags, observability metadata, and gap semantics.
Fixed-predictor long-duration use	Not supported as standalone flight capability	Add predictor-validity monitoring, recalibration policy, or bounded adaptive correction.
Adaptive correction	Supported for further prototype development	Require bounded updates, update gating, anomaly-state freeze, diagnostic visibility, and broader fault-family validation.
Beacon-style operational trust	Not yet supported	Require operations validation, false-alert tolerance assessment, fail-safe fallback rules, and operator acceptance testing.

The strongest evidence for continued development is embedded feasibility. The strongest limitation for flight readiness is predictor mismatch. A fixed residual detector can produce excessive false alerts when the nominal predictor becomes stale or mode-mismatched. Therefore, direct flight use without predictor-validity safeguards would risk producing an alert stream that operators cannot trust.

14.4. Limitations and Required Next Steps

The main limitations are validity and integration limitations, not basic implementation-feasibility limitations. The synthetic benchmark provides controlled event-level labels, but it does not establish real Delfi Twin fault statistics. The nominal thermal truth and predictor share simplified assumptions, so matched-predictor performance may be optimistic relative to real telemetry. The injected event families cover selected behaviours, not all possible faults, mixed events, mode transitions, or telemetry artifacts. FUNcube-1 provides useful real-telemetry stress evidence, but not labelled fault validation. Embedded replay verifies the detector core on target hardware, but not the complete flight-software data path, persistent storage, downlink scheduler, or flight-board power.

These limitations define the development path. The next phase should prioritize three activities.

Phase 1: Integration readiness

The first priority is to integrate the detector with the actual OBC telemetry task. This includes replacing UART replay with flight-like telemetry ingest, validating raw packet decoding and engineering-unit conversion, connecting telemetry-validity flags to the gap-handling logic, and confirming that the detector runs correctly under representative interrupt load, bus activity, storage, and downlink operations.

The embedded timing results should be repeated on the final target build with representative flight software running. Packet generation and storage should be measured in the final data path, not only in replay.

Phase 2: Operational validity

The second priority is to validate the predictor and detector under more realistic thermal and operational conditions. This includes hardware-in-the-loop thermal simulation, thermal-chamber testing if available, and replay of real or flight-like telemetry covering multiple operating modes.

Mode-aware operation should be added before any flight deployment. At minimum, the detector should distinguish Nominal, Safe, Boot, ADCS-active, communications-active, and heater-related operating states, or explicitly disable scoring when the context is outside its validity envelope.

Predictor-validity monitoring should also be added. This could include tracking residual baseline drift, common correction terms, local correction terms, residual scale changes, and persistent mismatch indicators. These values should be visible to operators rather than silently absorbed.

Phase 3: Trust and autonomy

The final priority is operator validation. Operators should review the alert fields, decoded text, false-alert tolerance, severity rules, and context-window requirements. This review should determine what information is needed for triage and what alert frequency is acceptable.

Only after these steps should beacon-style operations or autonomous downlink prioritization be evaluated. Beacon-style trust requires more than compact event packets. It requires evidence that the detector remains valid across modes, predictor drift, telemetry gaps, and nuisance artifacts, with fail-safe fallback when the detector is uncertain.

14.5. Chapter Summary

This chapter interpreted the prototype from an operational perspective. The results support continued development of the pipeline as an advisory on-board anomaly-awareness function. The prototype provides value through earlier event awareness, compact event reporting, bandwidth triage, and detector-faithful operator evidence. The embedded implementation also shows that MCU-class timing and memory feasibility are not the primary barriers.

Direct flight deployment is not yet recommended. The main deployment risk is predictor validity: residual evidence is meaningful only while the expected-temperature model remains aligned with the current operating context. Adaptive correction can reduce mismatch-driven alert burden, but it requires strict safeguards to avoid suppressing local fault evidence.

The appropriate next step is continued engineering development: integrate the detector with the flight telemetry path, validate timing under realistic system load, add mode-aware configuration, test against hardware-in-the-loop or thermal-chamber data, and validate alert usefulness with operators.

15

Conclusions

This thesis set out to determine whether an explainable, deployment-oriented, on-board temperature Anomaly Detection (AD) pipeline could be defined for Delfi Twin-class spacecraft and demonstrated on Microcontroller Unit (MCU)-class hardware. The work was motivated by a practical operational need: small satellites have limited downlink and operator attention, yet increasingly rich telemetry streams. A useful on-board AD function should therefore provide early awareness, compact event reporting, and operator-facing evidence, without exceeding on-board timing, memory, packet-size, or power constraints.

The thesis did not treat AD as an algorithm-selection problem alone. Instead, it treated it as a deployment problem. This required defining the mission role, telemetry scope, anomaly semantics, nominal thermal baseline, labelled benchmark, residual scoring method, event lifecycle, alert payload, embedded implementation, and operational readiness limits as one connected pipeline.

The main conclusion is that an explainable residual-to-event thermal AD pipeline is technically feasible on an STM32L4-class prototype, but is not yet flight-ready as a trusted autonomous capability. The embedded implementation demonstrates ample compute and memory margins, and the detector recovers alert-worthy events under matched-predictor assumptions. However, the evaluation also shows that predictor validity is the central deployment risk. If the expected-temperature model becomes stale, biased, or mode-mismatched, residual evidence can become misleading and produce persistent false-alert states. Continued development should therefore focus less on raw compute feasibility and more on predictor-validity monitoring, mode-aware calibration, guarded adaptation, flight-software integration, and operator validation.

Accordingly, this thesis supports prototype-level behaviour claims under controlled synthetic labels and STM32L4 replay; it does not validate flight performance, real-fault performance statistics, or autonomous operational trust.

15.1. Answers to the Research Questions

The research objective of this thesis was:

Research Objective

To design, implement, and evaluate a deployment-oriented, explainable, on-board temperature AD pipeline for Delfi Twin as an MCU-class case study, while developing a reusable workflow and evaluation protocol for assessing event-level thermal alerting in downlink-constrained space missions.

This section closes the thesis against the research objective by answering the six research questions defined in Chapter 5. Each answer is stated at the maturity level demonstrated in the thesis: a deployment-oriented prototype evaluated through controlled synthetic labels, baseline and ablation studies, STM32L4 replay, predictor-mismatch stress testing, and qualitative FUNcube-1 real-telemetry stress tests.

RQ-1: Operational role and requirements

How should on-board thermal AD be operationally formulated for a downlink-constrained mission, and what real-time, alerting, responsibility, and embedded constraints follow when this role is instantiated for Delfi Twin?

The detector should serve as an on-board advisory event-awareness function for temperature telemetry, not as an autonomous root-cause diagnosis or recovery system. This role is defined by the operational use case, responsibility allocation, timing budget, and embedded resource constraints developed in Section 3.1, Section 3.2, Section 3.3, and Section 3.4. The resulting requirements are compact event alerts, bounded on-board detection latency, explicit separation between spacecraft evidence reporting and ground-based diagnosis, and feasibility within STM32L4-class timing, memory, packet-size, and power margins.

RQ-2: Telemetry suitability and anomaly meaning

Which temperature telemetry channels and anomaly classes are suitable for a first deployment-oriented on-board AD prototype, and how should nominal thermal behaviour, telemetry pathologies, and physical thermal faults be distinguished?

Temperature telemetry is suitable as a first on-board AD target because it is physically meaningful, already available in housekeeping data, slowly varying enough for lightweight expected-behaviour modelling, and sensitive to both internal dissipation and environmental or illumination effects. This suitability and its limits are developed in Section 4.1, Section 4.2, Section 4.4, Section 4.5, and Section 4.7. The thesis therefore distinguishes nominal thermal variation, physical thermal deviations, telemetry-quality artifacts, benign nuisance behaviour, and loss of observability. Temperature telemetry can support early awareness and triage, but it does not by itself establish physical root cause.

RQ-3: Evaluation strategy without reliable real fault labels

How can a controlled and reproducible benchmark be constructed to evaluate event-level thermal AD when real spacecraft telemetry lacks reliable fault-level and alert-worthiness labels?

The thesis uses a layered evaluation strategy. Controlled synthetic labels provide the event-level information needed to compute alert-worthy recall, false-event burden, time-to-awareness, nuisance sensitivity, and event-fragmentation metrics. The synthetic benchmark construction is described in Chapter 7, and its interpretation boundaries are defined in Section 12.2. Because synthetic labels cannot establish real flight performance, the thesis also uses FUNcube-1 windows as qualitative real-telemetry stress tests in Chapter 13. These windows show whether the workflow remains inspectable on messy telemetry containing artifacts, contact gaps, contextual thermal variation, and apparent spin or illumination-imprinted structure. They are not used to compute real-data recall or precision.

FUNcube-1 provided morphology-level real-telemetry support for selected benchmark assumptions, including orbit-scale thermal structure, panel ripple, contact gaps, partial observability, and telemetry-quality artifacts, but it could not validate internal thermal fault labels because the required battery, MCU, and internal component temperature channels were unavailable.

RQ-4: Pipeline selection and alert formation

Which end-to-end AD pipeline is most appropriate for the selected telemetry, anomaly scope, and deployment constraints, and how should residual evidence be converted into bounded, explainable alerts?

The selected pipeline uses expected-temperature prediction, residual scoring, normalized evidence, instantaneous and cumulative decision logic, event lifecycle rules, and compact detector-faithful alert packets. The design rationale is developed in Chapter 8, Chapter 9, and Chapter 10. Empirical support comes from the decision-logic ablation and F1 adaptive-limits baseline, which show that simple thresholding or causal local adaptive limits do not provide the required recall, false-burden, and event-usability trade-off.

This pipeline complements rather than replaces conventional limit-based Fault Detection, Isolation and Recovery (FDIR). Hard limits remain necessary for deterministic protection, while the residual-to-event layer provides advisory evidence for context-dependent deviations that may occur within absolute operating bounds.

RQ-5: Embedded implementation and performance

Can the selected pipeline be implemented causally on STM32L4-class hardware while satisfying the timing, memory, packet-size, and resource constraints derived for the case-study mission?

The selected fixed and adaptive residual-to-event pipeline can be implemented on STM32L4-class hardware with large timing and memory margins. Embedded replay showed worst-case sample processing of approximately 12.8 ms, far below the 15 s telemetry cadence and 60 s alert-frame target. Node-wise adaptive correction added negligible timing overhead and modest memory cost. Targeted embedded mismatch replays confirmed that the adaptive path was functionally active on hardware.

These results support the embedded feasibility claim at prototype level. They do not prove full flight-software readiness, because final timing, memory, packet storage, bus interaction, and power behaviour still need to be measured in the integrated flight context.

RQ-6: Operational value and deployment readiness

What operational value does the prototype provide for downlink-constrained thermal anomaly awareness, what are its main limitations, and what further validation is required before flight deployment or autonomous beacon-style use?

The prototype provides mission value as an advisory residual-event generator for further engineering development and hardware-in-the-loop testing. Matched synthetic evaluation, ablation and baseline comparisons, embedded fixed/adaptive replay, targeted mismatch confirmation, and FUNcube stress tests support the conclusion that the pipeline is useful as a prototype-level anomaly-awareness function. These findings are synthesized in Section 12.10 and Section 12.11. However, the work does not demonstrate flight readiness. The principal no-go conditions are developed in Chapter 14: predictor validity, mode-aware calibration, guarded adaptation, raw telemetry integration, flight-software scheduling, downlink integration, and operator validation remain unresolved before direct flight or autonomous beacon-style use.

Together, these answers show that the thesis objective was achieved at the prototype and evaluation-framework level. The work demonstrates a credible path from mission need to embedded anomaly-alert implementation, while also identifying the conditions that must be satisfied before operational deployment. The main remaining limitation is not STM32 compute feasibility, but the operational governance of predictor validity, adaptation, telemetry context, and operator trust.

15.2. Contributions

This thesis contributed a deployment-oriented pathway for explainable on-board thermal anomaly alerting in downlink-constrained missions, instantiated through Delfi Twin as an MCU-class case study. The individual primitives used in the thesis, including residual scoring, cumulative evidence, persistence, hysteresis, detector-faithful alerting, and embedded timing measurement, are established techniques. The thesis contribution is their mission-specific selection, integration, implementation, evaluation, and claim-bounded interpretation as a coherent residual-to-alert workflow for event-level thermal anomaly awareness.

The three main contributions were:

Thesis Contributions

C-1: Deployment-oriented workflow for event-level thermal alerting.
The thesis defined an advisory early-awareness role for on-board temperature anomaly detection, including the fault management responsibility split, timing target, embedded constraints, telemetry scope, anomaly semantics, synthetic evaluation protocol, and interpretation limits needed for deployment-oriented assessment.

C-2: Explainable residual-to-event detector and compact alert packets.
The thesis selected and implemented a lightweight residual-based architecture that combines expected-temperature prediction, normalized residual evidence, instantaneous and cumulative evidence, event-lifecycle rules, and compact detector-faithful alert fields. The resulting alerts report event timing, affected channel, deviation magnitude, thermal direction, persistence, and trigger evidence without claiming on-board root-cause diagnosis.

C-3: Embedded verification, robustness testing, and deployment-readiness assessment.
The thesis implemented the pipeline on STM32L4-class hardware, verified timing, memory use, packet size, and resource feasibility, evaluated baseline and ablation cases, tested predictor-mismatch robustness, and applied the workflow to FUNcube-1 telemetry as qualitative real-telemetry stress cases. The assessment identified the conditions under which the prototype provides operational value, and the limitations that must be addressed before flight deployment or autonomous beacon-style use.

15.3. Final Synthesis

The thesis demonstrates that explainable on-board thermal anomaly alerts are technically feasible for a Delfi Twin-class spacecraft at the prototype level. The strongest evidence is the combination of controlled labelled evaluation, ablation and baseline justification, STM32 fixed/adaptive replay, and targeted embedded mismatch confirmation. The detector recovers alert-worthy events under matched-predictor assumptions, the selected event logic is justified against simpler alternatives, and the adaptive predictor can run and correct representative mismatch on STM32 hardware.

A key interpretation of this result is that the prototype addresses a complementary role to conventional limit-based FDIR: hard temperature limits remain necessary for protection, but expected-temperature residuals and lifecycle logic provide earlier context-aware evidence for deviations that may remain within absolute operating bounds.

The embedded evidence is particularly important. The prototype shows that MCU-class compute is not the primary barrier for this pipeline. The ordinary predictor and detector path executes far below the telemetry cadence, and the memory footprint is small relative to the target resource envelope. The compact event-packet format also supports the operational concept of prioritizing alert summaries and associated context windows rather than downlinking continuous full-resolution telemetry.

However, the thesis also shows that successful matched-predictor detection is not sufficient for operational trust. The residual detector depends on the validity of the expected-temperature predictor. When the predictor becomes biased or stale, cumulative residual evidence can produce persistent false-alert states. This is the central deployment risk identified by the work. It means that future development should prioritize predictor-validity monitoring, mode-aware calibration, and bounded adaptation safeguards before considering autonomous use.

The work does not demonstrate flight readiness. The synthetic labels are controlled rather than observed flight faults, the FUNcube examples are unlabelled stress tests, the embedded evaluation uses replay rather than integrated flight software, and adaptation is only validated as a bounded node-wise prototype. Therefore, the prototype should be viewed as a credible advisory anomaly-awareness prototype and evaluation framework, not as a flight-qualified autonomous FDIR system.

15.4. Recommendations

This section gives the final recommendations for continuing the work. The objective is to leave a concise path from thesis prototype to mission implementation.

1. The pipeline should be continued as an *advisory on-board anomaly-awareness function*, not as an autonomous response system. Its near-term value is earlier awareness, compact event reporting, and context-window prioritization.
2. Future work should prioritize *flight-software integration* before adding more algorithmic complexity. The detector should be connected to the actual OBC telemetry task, raw telemetry decoding, validity flags, storage, and downlink pathway. Timing should be remeasured under representative interrupt, bus, and storage load.
3. The predictor and detector should become *mode-aware*. Normal thermal behaviour may differ across Boot, Safe, Nominal, ADCS-active, communications-active, payload-active, and heater-related states. Mode-conditioned templates, thresholds, or scoring policies are likely required before flight deployment.
4. The embedded adaptive predictor should be retained as a prototype mitigation, but its update policy should remain conservative. Future work should define when adaptation is allowed, frozen, reset, or surfaced to operators, and should validate those policies across modes and multi-node scenarios.
5. Temperature-only detection should be extended with *voltage, current, and operational context*. Temperature is useful for early awareness, but voltage/current context would improve diagnostic value and help distinguish thermal faults from telemetry-quality issues, mode changes, or power-system behaviour.
6. *Operator-facing validation* is needed before autonomous trust. Operators should review alert wording, evidence fields, false-alert tolerance, severity rules, and context-window requirements. Beacon-style operation or autonomous downlink suppression should only be considered after this validation.

15.5. Closing Statement

The main lesson of this thesis is that deployable spacecraft AD is a system design problem, not only an algorithm-selection problem. A useful on-board detector must be built around the mission role, telemetry observability, anomaly semantics, ground-truth protocol, decision logic, alert interface, and embedded constraints.

Under that framing, this thesis provides evidence that explainable on-board thermal anomaly alerts are feasible for Delfi Twin-class spacecraft. The STM32 prototype demonstrates that the core residual-to-alert pathway can run within MCU-class constraints and produce compact, interpretable event packets.

The remaining challenge is not whether the microcontroller can execute the algorithm, but whether predictor context, adaptation policy, flight-software integration, and operator workflow can be validated well enough for operational trust.

References

[1] Mattia Antonini et al. “An Adaptable and Unsupervised TinyML Anomaly Detection System for Extreme Industrial Environments”. en. In: *Sensors* 23.4 (Jan. 2023), p. 2344. ISSN: 1424-8220. DOI: 10.3390/s23042344. URL: https://www.mdpi.com/1424-8220/23/4/2344 (visited on 06/01/2026).

[2] A Anvari, Foad Farhani, and Keyvan Niaki. “Comparative Study on Space Qualified Paints Used for Thermal Control of a Small Satellite”. In: *Iranian Journal of Chemical Engineering* 6 (Jan. 2009).

[3] *AO-73 (FUNcube-1) – AMSAT*. URL: https://www.amsat.org/two-way-satellites/ao-73-funcube-1/ (visited on 05/24/2026).

[4] Alessandro Battegazzore. “A Compact Radio Beacon and Antenna Deployer Design”. MSc thesis. Delft, Netherlands: Delft University of Technology, 2025.

[5] Ullas Bhat. “Neural-Network Based Thermal Modeling of Small Satellites”. MSc thesis. Delft University of Technology, 2023. URL: https://repository.tudelft.nl/record/uuid:b56b443a-3097-46b2-ac23-9116d15628bd (visited on 09/16/2025).

[6] Marie Bieber et al. “Generic Diagnostic Framework for Anomaly Detection—Application in Satellite and Spacecraft Systems”. In: *Aerospace* 10.8 (Aug. 2023), p. 673. ISSN: 2226-4310. DOI: 10.3390/aerospace10080673. URL: https://www.mdpi.com/2226-4310/10/8/673 (visited on 10/07/2025).

[7] Danilo Cappellone et al. “On-Board Satellite Telemetry Forecasting with RNN on RISC-V Based Multicore Processor”. In: *33rd IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, DFT 2020* (2020). Ed. by Luigi Dilillo, Mihalis Psarakis, and Taniya Siddiqua. DOI: 10.1109/DFT50435.2020.9250796. URL: http://www.scopus.com/inward/record.url?scp=85097656469&partnerID=8YFLogxK (visited on 04/01/2026).

[8] Lorenzo Pasqualetto Cassinis. “Modeling and Analysis of Delfi-C3 Telemetry”. PhD thesis. Delft University of Technology, July 2017.

[9] Varun Chandola, Arindam Banerjee, and Vipin Kumar. *Anomaly Detection : A Survey*. Technical Report TR 07-017. Minneapolis, Minnesota, USA: University of Minnesota, Aug. 2007, p. 74.

[10] Varun Chandola, Arindam Banerjee, and Vipin Kumar. “Anomaly Detection: A Survey”. In: *ACM Computing Surveys* 41.3 (July 2009), pp. 1–58. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/1541880.1541882. URL: https://dl.acm.org/doi/10.1145/1541880.1541882 (visited on 10/08/2025).

[11] Sara Cuéllar et al. “Explainable Anomaly Detection in Spacecraft Telemetry”. In: *Engineering Applications of Artificial Intelligence* 133 (July 1, 2024), p. 108083. ISSN: 0952-1976. DOI: 10.1016/j.engappai.2024.108083. URL: https://www.sciencedirect.com/science/article/pii/S0952197624002410 (visited on 09/12/2025).

[12] *Datasheet - MSP432P411x*. Texas Instruments, Dec. 2017.

[13] *Datasheet - STM32L476xx*. ST Microelectronics, July 2024.

[14] *Datasheet - STM32L496xx*. ST Microelectronics, Oct. 2025.

[15] Gabriele De Canio, Krzysztof Kotowski, and Christoph Haskamp. *ESA Anomaly Dataset*. European Space Agency, Apr. 17, 2025. DOI: 10.5281/zenodo.15237121. URL: https://zenodo.org/records/15237121 (visited on 10/16/2025).

[16] *Designing a CAN Network*. URL: https://www.can-cia.org/can-knowledge/designing-a-can-network (visited on 11/11/2025).

[17] Tristan Dijkstra. “Machine Learning-based Anomaly Detection in XMM-Newton Telemetry Data”. MSc thesis. Delft University of Technology, 2025.

[18] M. A. G. Duff, N. D. F. Campbell, and M. J. Ehrhardt. “Regularising Inverse Problems with Generative Machine Learning Models”. In: *Journal of Mathematical Imaging and Vision* 66.1 (Jan. 2024), pp. 37–56. ISSN: 0924-9907, 1573-7683. DOI: 10.1007/s10851-023-01162-x. URL: https://link.springer.com/10.1007/s10851-023-01162-x (visited on 03/24/2026).

[19] ESA Requirements and Standards Division. *Space product assurance: Failure modes, effects (and criticality) analysis (FMEA/FMECA)*. Standard ECSS-Q-ST-30-02C. Reconfirmed by the ECSS Technical Authority on 14 June 2023. European Cooperation for Space Standardization, Mar. 2009. URL: https://ecss.nl/standard/ecss-q-st-30-02c-failure-modes-effects-and-criticality-analysis-fmeafmeca/ (visited on 05/21/2026).

[20] *Fault Management Handbook*. NASA Technical Handbook NASA-HDBK-1002. Version Draft 2. Apr. 2, 2012.

[21] Asma Fejjari et al. “A Review of Anomaly Detection in Spacecraft Telemetry Data”. In: *Applied Sciences* 15.10 (Jan. 2025), p. 5653. ISSN: 2076-3417. DOI: 10.3390/app15105653. URL: https://www.mdpi.com/2076-3417/15/10/5653 (visited on 10/10/2025).

[22] *FUNcube-1 Whole Orbit Data · AMSAT-UK Data Warehouse*. URL: http://data.amsat-uk.org/ui/fc1-fm/wod (visited on 05/24/2026).

[23] Eran Gal and Sivan Toledo. “Algorithms and data structures for flash memories”. In: *ACM Computing Surveys* 37.2 (June 2005), pp. 138–163. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/1089733.1089735. URL: https://dl.acm.org/doi/10.1145/1089733.1089735 (visited on 01/11/2026).

[24] Alejandro Garzón et al. “Effect of beta angle and contact conductances on the temperature distribution of a 3U CubeSat”. In: *Thermal Science and Engineering Progress* 29 (Mar. 2022), p. 101183. ISSN: 2451-9049. DOI: 10.1016/j.tsep.2021.101183. URL: https://www.sciencedirect.com/science/article/pii/S2451904921003425 (visited on 11/18/2025).

[25] David Gilmore. *Spacecraft Thermal Control Handbook*. Second Edition. Vol. Volume I: Fundamental Technologies. Washington, DC: American Institute of Aeronautics and Astronautics, Inc., Dec. 2002. ISBN: 978-1-884989-11-7. DOI: 10.2514/4.989117. URL: https://arc.aiaa.org/doi/book/10.2514/4.989117 (visited on 11/19/2025).

[26] Gianluca Giuffrida et al. “The Φ-Sat-1 Mission: The First On-Board Deep Neural Network Demonstrator for Satellite Earth Observation”. In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), pp. 1–14. ISSN: 1558-0644. DOI: 10.1109/TGRS.2021.3125567. URL: https://ieeexplore.ieee.org/document/9600851 (visited on 10/07/2025).

[27] Lars Herrmann et al. “Unmasking overestimation: a re-evaluation of deep anomaly detection in spacecraft telemetry”. In: *CEAS Space Journal* 16.2 (Mar. 2024), pp. 225–237. ISSN: 1868-2510. DOI: 10.1007/s12567-023-00529-5. URL: https://doi.org/10.1007/s12567-023-00529-5 (visited on 05/25/2026).

[28] T. D. Hirs. “Physics-Informed Neural Networks for Aerospace Applications”. MSc thesis. Delft University of Technology, 2025. URL: https://repository.tudelft.nl/record/uuid:76d93202-df29-46b4-ba62-452cc5532b4d.

[29] Ross Horne et al. “Anomaly Detection Using Deep Learning Respecting the Resources on Board a CubeSat”. In: *Journal of Aerospace Information Systems* 20.12 (Dec. 2023), pp. 859–872. ISSN: 1940-3151, 2327-3097. DOI: 10.2514/1.I011232. URL: https://arc.aiaa.org/doi/10.2514/1.I011232 (visited on 10/29/2025).

[30] Donald A. Jaworske and Sara E. Kline. *Review of End-of-Life Thermal Control Coating Performance*. Tech. rep. NASA/TM-2008-215173. NTRS Author Affiliations: NASA Glenn Research Center, Akron Univ. NTRS Document ID: 20080018585 NTRS Research Center: Glenn Research Center (GRC). Apr. 2008. URL: https://ntrs.nasa.gov/citations/20080018585 (visited on 12/04/2025).

[31] Weihua Jin et al. “Detecting Anomalies of Satellite Power Subsystem via Stage-Training Denoising Autoencoders”. In: *Sensors (Basel, Switzerland)* 19.14 (July 22, 2019), p. 3216. ISSN: 1424-8220. DOI: 10.3390/s19143216. PMID: 31336565. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC6679529/ (visited on 11/13/2025).

[32] Beomsik Kim and Hoeseok Yang. “Reliability Optimization of Real-Time Satellite Embedded System Under Temperature Variations”. In: *IEEE Access* 8 (2020), pp. 224549–224564. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3044044. URL: https://ieeexplore.ieee.org/document/9291428/ (visited on 11/14/2025).

[33] Vaclav Knap, Lars Kjeldgaard Vestergaard, and Daniel-Ioan Stroe. “A Review of Battery Technology in CubeSats and Small Satellite Solutions”. In: *Energies* 13.16 (Jan. 2020). Publisher: Multidisciplinary Digital Publishing Institute, p. 4097. ISSN: 1996-1073. DOI: 10.3390/en13164097. URL: https://www.mdpi.com/1996-1073/13/16/4097 (visited on 12/03/2025).

[34] Ana Kostovska et al. *GalaxAI: Machine learning toolbox for interpretable analysis of spacecraft telemetry data*. arXiv:2108.01407 [cs]. Aug. 2021. DOI: 10.48550/arXiv.2108.01407. URL: http://arxiv.org/abs/2108.01407 (visited on 04/01/2026).

[35] Krzysztof Kotowski et al. *European Space Agency Benchmark for Anomaly Detection in Satellite Telemetry*. Aug. 17, 2025. DOI: 10.48550/arXiv.2406.17826. arXiv: 2406.17826 [cs]. URL: http://arxiv.org/abs/2406.17826 (visited on 10/16/2025). Pre-published.

[36] Theodora Kourti and John F. MacGregor. “Process analysis, monitoring and diagnosis, using multivariate projection methods”. In: *Chemometrics and Intelligent Laboratory Systems* 28.1 (Apr. 1995), pp. 3–21. ISSN: 0169-7439. DOI: 10.1016/0169-7439(95)80036-9. URL: https://www.sciencedirect.com/science/article/pii/0169743995800369 (visited on 01/23/2026).

[37] Zhong Li, Yuxuan Zhu, and Matthijs Van Leeuwen. “A Survey on Explainable Anomaly Detection”. In: *ACM Trans. Knowl. Discov. Data* 18.1 (Sept. 2023), 23:1–23:54. ISSN: 1556-4681. DOI: 10.1145/3609333. URL: https://dl.acm.org/doi/10.1145/3609333 (visited on 01/16/2026).

[38] Kaitlin Liles and Ruth Amundsen. *NASA Passive Thermal Control Engineering Guidebook*. Tech. rep. Version: 5.1. NTRS Author Affiliations: Langley Research Center NTRS Document ID: 20220006584 NTRS Research Center: Langley Research Center (LaRC). Dec. 2024. URL: https://ntrs.nasa.gov/citations/20220006584 (visited on 12/03/2025).

[39] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. “Isolation Forest”. In: *2008 Eighth IEEE International Conference on Data Mining*. ISSN: 2374-8486. Dec. 2008, pp. 413–422. DOI: 10.1109/ICDM.2008.17. URL: https://ieeexplore.ieee.org/document/4781136 (visited on 01/26/2026).

[40] Tianyu Liu et al. “Degradation modeling of satellite thermal control coatings in a low earth orbit environment”. In: *Solar Energy* 139 (Dec. 2016), pp. 467–474. ISSN: 0038-092X. DOI: 10.1016/j.solener.2016.10.031. URL: https://www.sciencedirect.com/science/article/pii/S0038092X16304960 (visited on 12/09/2025).

[41] Chenhui Luan et al. “Anomaly Detection for Slowly Varying Analog Telemetry Data of Spacecraft Based on Autoencoder”. In: *2024 3rd International Symposium on Sensor Technology and Control (ISSTC)* (Oct. 25, 2024), pp. 307–311. DOI: 10.1109/ISSTC63573.2024.10824188. URL: https://ieeexplore.ieee.org/document/10824188/ (visited on 11/13/2025).

[42] Ryan Mackey and Igor Kulikov. “Forecasting Spacecraft Telemetry Using Modified Physical Predictions”. In: *Annual Conference of the PHM Society* 2 (Oct. 2010). DOI: 10.36001/phmconf.2010.v2i1.1803.

[43] *Management of Alarm Systems for the Process Industries*. Mar. 2016.

[44] James Paul Mason et al. “MinXSS-2 CubeSat mission overview: Improvements from the successful MinXSS-1 mission”. In: *Advances in Space Research*. Advances in Small Satellites for Space Science 66.1 (July 2020), pp. 3–9. ISSN: 0273-1177. DOI: 10.1016/j.asr.2019.02.011. URL: https://www.sciencedirect.com/science/article/pii/S0273117719301152 (visited on 11/18/2025).

[45] Rudolf Maununen. “Neural Network for Temperature Monitoring Deployed on a Low-power CubeSat Onboard Computer”. MSc thesis. Delft University of Technology, 2023. URL: https://repository.tudelft.nl/record/uuid:e8f11bd7-4441-4866-bdae-aaf9bab6744c (visited on 09/15/2025).

[46] Amirhossein Moallemi et al. “Exploring Scalable, Distributed Real-Time Anomaly Detection for Bridge Health Monitoring”. en. In: *IEEE Internet of Things Journal* 9.18 (Sept. 2022), pp. 17660–17674. ISSN: 2327-4662, 2372-2541. DOI: 10.1109/JIOT.2022.3157532. URL: https://ieeexplore.ieee.org/document/9729869/ (visited on 06/01/2026).

[47] Edemar Morsch Filho et al. “A comprehensive attitude formulation with spin for numerical model of irradiance for CubeSats and Picosats”. In: *Applied Thermal Engineering* 168 (Mar. 2020), p. 114859. ISSN: 1359-4311. DOI: 10.1016/j.applthermaleng.2019.114859. URL: https://www.sciencedirect.com/science/article/pii/S1359431119333459 (visited on 11/19/2025).

[48] Ryan Mukai et al. “MSL Telecom Automated Anomaly Detection”. In: *2020 IEEE Aerospace Conference*. 2020 IEEE Aerospace Conference. Mar. 2020, pp. 1–6. DOI: 10.1109/AERO47225.2020.9172573. URL: https://ieeexplore.ieee.org/document/9172573/ (visited on 10/07/2025).

[49] Tahir Munir et al. “Effect of measurement uncertainty on combined quality control charts”. In: *Computers & Industrial Engineering* 175 (Jan. 2023), p. 108900. ISSN: 0360-8352. DOI: 10.1016/j.cie.2022.108900. URL: https://www.sciencedirect.com/science/article/pii/S0360835222008889 (visited on 01/29/2026).

[50] *NASA Anomaly Detection Dataset SMAP & MSL*. URL: https://www.kaggle.com/datasets/patrickfleith/nasa-anomaly-detection-dataset-smap-msl (visited on 04/17/2026).

[51] Andrei Paleyes, Raoul-Gabriel Urma, and Neil D. Lawrence. “Challenges in Deploying Machine Learning: A Survey of Case Studies”. In: *ACM Comput. Surv.* 55.6 (Dec. 2022), 114:1–114:29. ISSN: 0360-0300. DOI: 10.1145/3533378. URL: https://dl.acm.org/doi/10.1145/3533378 (visited on 02/27/2026).

[52] Dawei Pan et al. “Anomaly Detection for Satellite Power Subsystem with Associated Rules Based on Kernel Principal Component Analysis”. In: *Microelectronics Reliability*. Proceedings of the 26th European Symposium on Reliability of Electron Devices, Failure Physics and Analysis 55.9 (Aug. 1, 2015), pp. 2082–2086. ISSN: 0026-2714. DOI: 10.1016/j.microrel.2015.07.010. URL: https://www.sciencedirect.com/science/article/pii/S0026271415300949 (visited on 11/13/2025).

[53] Eric Pesola et al. “A Hybrid Model-Based and Data-Driven Framework for Automated Spacecraft Fault Detection”. In: *Annual Conference of the PHM Society* 15.1 (Oct. 26, 2023). ISSN: 2325-0178. DOI: 10.36001/phmconf.2023.v15i1.3461. URL: https://papers.phmsociety.org/index.php/phmconf/article/view/3461 (visited on 10/07/2025).

[54] Silvana Radu et al. “Delfi-PQ: The First Pocketqube of Delft University of Technology”. In: *Proceedings of 69th International Astronautical Congress*. Bremen, Germany: International Astronautical Federation, IAF, 2018.

[55] Jorge Reyes-Marambio et al. “A fractal time thermal model for predicting the surface temperature of air-cooled cylindrical Li-ion cells based on experimental measurements”. In: *Journal of Power Sources* 306 (Feb. 2016), pp. 636–645. ISSN: 0378-7753. DOI: 10.1016/j.jpowsour.2015.12.037. URL: https://www.sciencedirect.com/science/article/pii/S0378775315306509 (visited on 05/09/2026).

[56] Paul D. Rosero-Montalvo et al. “Hybrid Anomaly Detection Model on Trusted IoT Devices”. In: *IEEE Internet of Things Journal* 10.12 (June 2023), pp. 10959–10969. ISSN: 2327-4662. DOI: 10.1109/JIOT.2023.3243037. URL: https://ieeexplore.ieee.org/document/10039052/ (visited on 06/01/2026).

[57] *Rules for the Design, Development, Verification, and Operation of Flight Systems*. NASA Goddard Technical Standard GSFC-STD-1000. Version Revision I. Greenbelt, Maryland, Aug. 19, 2025. 106 pp. URL: https://standards.nasa.gov/system/files/tmp/GSFC-STD-1000RevI_Approved_0.pdf (visited on 11/11/2025).

[58] Bogdan Ruszczak et al. *The OPS-SAT Benchmark for Detecting Anomalies in Satellite Telemetry*. Version 1. June 29, 2024. DOI: 10.48550/arXiv.2407.04730. arXiv: 2407.04730 [eess]. URL: http://arxiv.org/abs/2407.04730 (visited on 10/20/2025). Pre-published.

[59] Robert A. Sayer et al. "Thermal Contact Conductance of Radiation-Aged Thermal Interface Materials for Space Applications". In: *Proceedings of the ASME 2013 Summer Heat Transfer Conference*. Minneapolis, Minnesota, USA: American Society of Mechanical Engineers, July 2013, V003T10A002. ISBN: 978-0-7918-5549-2. DOI: 10.1115/HT2013-17408. URL: https://asmedigitalcollection.asme.org/HT/proceedings/HT2013/55492/Minneapolis,%20Minnesota,%20USA/242556 (visited on 12/09/2025).

[60] Clemens Schefels, Leonard Schlag, and Kathrin Helmsauer. "Synthetic satellite telemetry data for machine learning". In: *CEAS Space Journal* 17.5 (Sept. 2025), pp. 863–875. ISSN: 1868-2510. DOI: 10.1007/s12567-024-00589-1. URL: https://doi.org/10.1007/s12567-024-00589-1 (visited on 11/26/2025).

[61] Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. "Anomaly detection in time series: a comprehensive evaluation". In: *Proc. VLDB Endow.* 15.9 (May 2022), pp. 1779–1797. ISSN: 2150-8097. DOI: 10.14778/3538598.3538602. URL: https://dl.acm.org/doi/10.14778/3538598.3538602 (visited on 03/05/2026).

[62] Bernhard Schölkopf et al. "Support Vector Method for Novelty Detection". In: *Advances in Neural Information Processing Systems*. Vol. 12. MIT Press, 1999. URL: https://proceedings.neurips.cc/paper_files/paper/1999/hash/8725fb777f25776ffa9076e44fcfd776-Abstract.html (visited on 01/26/2026).

[63] Jonathon Shlens. *A Tutorial on Principal Component Analysis*. San Diego La Jolla, CA, Dec. 2005.

[64] Stefano Speretta, R.K. van der Zwaard, and M.S. Uludag. "Autonomous Formation Flying in the Traffic". In: *Proceedings Small Satellites for Earth Observation*. Vol. Article DLR-IAA 2025-137 DLR. 2025, pp. 1–8.

[65] Stefano Speretta et al. "Command and Data Handling Systems". In: *Next Generation CubeSats and SmallSats: Enabling Technologies, Missions, and Markets*. Elsevier, 2023, pp. 369–399. ISBN: 978-0-12-824541-5. DOI: 10.1016/B978-0-12-824541-5.00012-1. URL: https://linkinghub.elsevier.com/retrieve/pii/B9780128245415000121 (visited on 11/10/2025).

[66] Bo Sun et al. "A Review of Fault Detection and Diagnosis of Satellite Power Subsystem". In: *Wireless and Satellite Systems*. Ed. by Qihui Wu, Kanglian Zhao, and Xiaojin Ding. Vol. 358. Cham: Springer International Publishing, 2021, pp. 164–174. DOI: 10.1007/978-3-030-69072-4_14. URL: http://link.springer.com/10.1007/978-3-030-69072-4_14 (visited on 11/13/2025).

[67] David M.J Tax and Robert P.W Duin. "Support vector domain description". In: *Pattern Recognition Letters* 20.11-13 (Nov. 1999), pp. 1191–1199. ISSN: 01678655. DOI: 10.1016/S0167-8655(99)00087-2. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167865599000872 (visited on 01/26/2026).

[68] Jan Thoemel et al. "Lean Demonstration of On-Board Thermal Anomaly Detection Using Machine Learning". en. In: *Aerospace* 11.7 (July 2024), p. 523. ISSN: 2226-4310. DOI: 10.3390/aerospace11070523. URL: https://www.mdpi.com/2226-4310/11/7/523 (visited on 06/01/2026).

[69] Cristopher Castro Traba. *ML Models on the MSP432P401R*. Delft University of Technology, 2024.

[70] Mehmet Sevket Uludag and Stefano Speretta. "Delfi-PQ: In-Orbit Performance and Lessons Learned in Developing a 3P PocketQube". In: Sydney, Australia, Oct. 2025.

[71] Yunhai Wang et al. "A Solar Array Temperature Multivariate Trend Forecasting Method Based on the CA-PatchTST Model". In: *Sensors (Basel, Switzerland)* 25.23 (Nov. 2025), p. 7199. ISSN: 1424-8220. DOI: 10.3390/s25237199. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC12694205/ (visited on 04/01/2026).

[72] Sasha V Weston et al. "Thermal Control". In: *State-of-the-Art Small Spacecraft Technology*. NASA, Feb. 2025, pp. 205–225.

[73] E Jay Wyatt et al. “Emerging Techniques for Deep Space Cubesat Operations”. In: Interplanetary CubeSat Workshop. Oxford, United Kingdom, May 25, 2016.

[74] Jingyan Xie and Yun-Ze Li. “Telemetry-Driven Physics-Informed Prediction for On-Orbit Thermal–Electrical Performance of Solar Panels in Satellite Power System”. In: *Aerospace Science and Technology* (Mar. 2026), p. 112268. ISSN: 1270-9638. DOI: 10.1016/j.ast.2026.112268. URL: https://www.sciencedirect.com/science/article/pii/S1270963826006486 (visited on 04/01/2026).

A

Deferred Thermal Fault Parameterization

This appendix summarizes the supporting analysis for thermal fault families that were identified as deployment-relevant in Chapter 4, but not injected as labelled event classes in the training, validation, or evaluation datasets. These families are still relevant because they can alter the expected thermal behaviour seen by a residual-based detector. In particular, **SPIN** motivates future attitude-aware thermal scenarios, while **DEG** and **COND** motivate predictor-mismatch and temperature-bias robustness tests.

A.1. SPIN: Spin, Attitude, and Illumination Effects

The thermal state of a small spacecraft depends on how long each external face sees the Sun, Earth, or deep space. Changes in attitude, spin rate, or precession therefore change the distribution of absorbed heat across the spacecraft. Low-spin or poor attitude pointing can produce large hot-cold gradients and orbit-scale swings on external faces, whereas high-speed spin tends to reduce those swings and equalize temperature gradients between opposite faces [47].

For Delfi Twin, spin and attitude anomalies are treated as rare Attitude Determination and Control System (ADCS)-related faults with thermal consequences, rather than as primary thermal failures. The nominal attitude concept assumes a high-spin regime, so large changes in spin rate or prolonged attitude misalignment are not expected during nominal operations. If they occur, their thermal signature would be expected first in the external panel channels, and only later in internal nodes after conduction and thermal inertia have filtered the disturbance.

The expected observable effects are:

- changes in orbit-scale panel amplitude;
- increased or decreased gradients between opposing faces;
- changes in spin-induced ripple amplitude or period;
- delayed and smaller responses in internal nodes.

This family was not implemented as a labelled event class in the final benchmark because a credible injection requires coupled attitude, illumination, and thermal modelling. A simplified amplitude or phase perturbation could be useful for sensitivity testing, but would not provide a sufficiently grounded fault label for the main event-recovery benchmark.

A.2. DEG: Surface and Coating Degradation

Surface and coating degradation represents slow changes in thermo-optical properties over mission life. External surfaces can change due to Ultraviolet (UV) exposure, atomic oxygen, contamination, and outgassing [30]. These changes can alter absorbed solar energy and emitted thermal radiation, producing slow changes in orbit-mean temperature and possibly orbit amplitude.

A.2.1. Thermo-optical temperature sensitivity

A simplified spacecraft thermal balance can be written as

$$Q_{\text{internal}} + Q_{\text{solar}} + Q_{\text{albedo}} + Q_{\text{infrared}} = Q_{\text{radiative}}. \tag{A.1}$$

For a simple flat-plate approximation with negligible internal heat and dominant direct solar input,

$$Q_{\text{solar}} \approx Q_{\text{radiative}} \quad \Rightarrow \quad \alpha_S S \approx \varepsilon_{IR} \sigma T^4, \tag{A.2}$$

where α_S is solar absorptance, S is solar irradiance, ε_{IR} is infrared emissivity, and σ is the Stefan-Boltzmann constant. Solving for temperature gives

$$T \approx \left(\frac{S}{\sigma} \frac{\alpha_S}{\varepsilon_{IR}} \right)^{1/4}. \tag{A.3}$$

Thus, under this idealized radiative-only model,

$$T \propto \left(\frac{\alpha}{\varepsilon} \right)^{1/4}. \tag{A.4}$$

In practice, studies of common Low Earth Orbit (LEO) thermal-control coatings report that emissivity changes are relatively small, while solar absorptance increases more substantially [30, 40]. The analysis therefore focuses on solar-absorptance ageing.

A.2.2. Solar absorptance ageing model

Thermal-control coatings often degrade rapidly early in mission life and then approach a slower, saturating trend. This behaviour is visible in the Z-93P absorptance data shown in Figure A.1.

The extracted coating data used to define an order-of-magnitude end-of-life trend are summarized in Table A.1. Across the selected materials, the cumulative absorptance increase over approximately five years is roughly 30 % to 40 %. This thesis adopts a representative End of Life (EOL) increase of 35 %.

The dimensionless absorptance-ageing factor is defined as

$$f_\alpha(t) = \frac{\alpha(t)}{\alpha_0}, \tag{A.5}$$

with $f_\alpha(0) = 1$ and $f_\alpha(T_{\text{mission}}) \approx 1.35$. A normalized saturating exponential is used to capture the rapid early increase and later saturation:

$$f_\alpha(t) = 1 + (f_{\text{EOL}} - 1) \frac{1 - e^{-t/\tau}}{1 - e^{-T_{\text{mission}}/\tau}}. \tag{A.6}$$

Based on the extracted coating data, $\tau = 2\,\text{yr}$ is selected as a representative time constant. Therefore,

$$f_\alpha(t) = 1 + 0.35 \frac{1 - e^{-t/2}}{1 - e^{-T_{\text{mission}}/2}}. \tag{A.7}$$

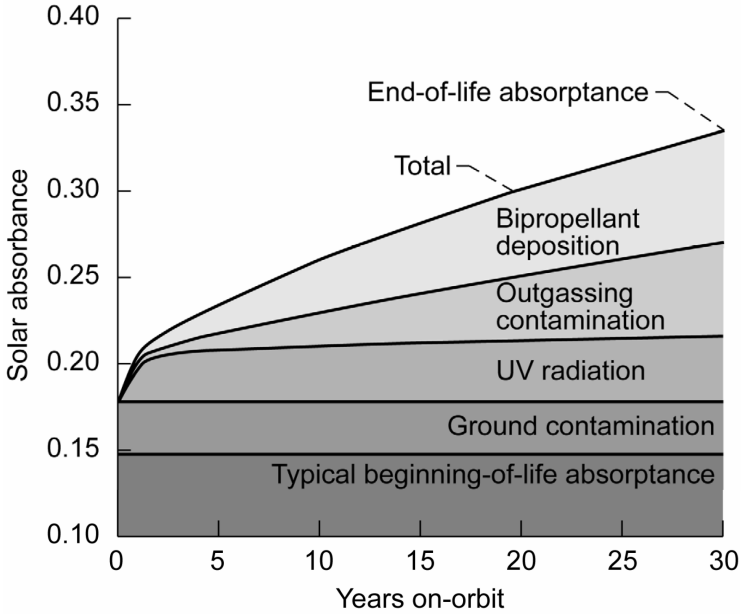


Figure A.1: Solar-absorptance increase of Z-93P on the International Space Station, with individual degradation contributions identified (from [30]).

Table A.1: Cumulative and annual growth of solar absorptance for selected thermal-control coatings in LEO, relative to beginning-of-life values.

Material	Year	Increase in α		Relative cumulative fraction [-]
		Cumulative [%]	Increment [%-pts/yr]	
Z93	1	14.78	14.78	0.48
	2	20.72	5.94	0.67
	3	24.56	3.84	0.79
	4	27.82	3.26	0.90
	5	30.95	3.13	1.00
S781	1	13.87	13.87	0.35
	2	26.24	12.37	0.67
	3	32.20	5.96	0.82
	4	36.20	4.00	0.92
	5	39.45	3.25	1.00
F46	1	13.82	13.82	0.41
	2	23.19	9.37	0.69
	3	30.83	7.64	0.92
	4	33.47	2.64	1.00
	5	–	–	–

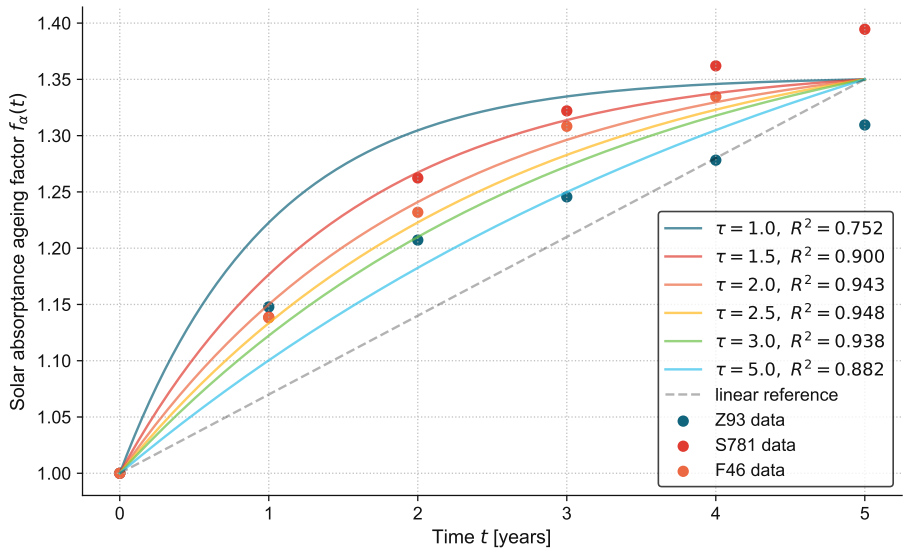


Figure A.2: Saturating exponential fit for the solar-absorptance ageing factor $f_\alpha(t)$ over a 5 yr mission, normalized to $f_\alpha(0) = 1$ and $f_\alpha(T) = 1.35$.

A.2.3. Temperature-magnitude interpretation

The radiative-only relationship gives the upper-bound scaling

$$\frac{T(t)}{T_0} \approx f_\alpha^{1/4}. \tag{A.8}$$

For $f_\alpha = 1.35$, this gives $f_\alpha^{1/4} \approx 1.077$, or a naive 7 % to 8 % temperature increase. However, real satellites are conductively coupled, and external-surface changes are damped by internal thermal mass and heat paths. Based on the small-satellite results of Anvari et al. [2], this work adopts the approximate linearized mapping

$$\Delta T_{\text{ext}} \approx (0.08\text{–}0.12)\,^\circ\text{C per 1\% increase in } \alpha, \qquad \Delta T_{\text{int}} \approx 0.6\text{–}0.7 \cdot \Delta T_{\text{ext}}. \tag{A.9}$$

For a representative 35 % absorptance increase over five years, this corresponds to approximately 3 °C to 4 °C drift on external panels and 2 °C to 3 °C on internal nodes. In this thesis, this magnitude is not used as a labelled event injection. Instead, it motivates the need to test residual-detector sensitivity to slow predictor bias and model-validity drift.

A.3. COND: Thermal-Interface and Conductance Degradation

Thermal-interface degradation represents changes in conductive coupling between spacecraft components. Thermal Interface Materials (TIMs), adhesives, and structural contacts can degrade under UV, radiation, atomic oxygen, thermal cycling, and outgassing. The resulting increase in thermal resistance changes the heat flow between components, producing altered thermal gradients, time constants, and orbit-amplitude coupling [59].

A.3.1. From contact resistance to conductance

Thermal contact resistance can be written as [25]

$$R = \frac{A\Delta T}{Q}, \tag{A.10}$$

where A is contact area, ΔT is the temperature difference, and Q is heat flow. Contact conductance is the inverse quantity,

$$G = \frac{1}{R}. \tag{A.11}$$

If thermal resistance increases by a factor f, the conductance decreases by the same factor:

$$R_{\text{new}} = fR_0 \quad \Rightarrow \quad G_{\text{new}} = \frac{1}{R_{\text{new}}} = \frac{G_0}{f}. \tag{A.12}$$

Sayer et al. [59] report substantial increases in contact resistance for space-relevant TIMs. Interpreted as conductance degradation, these results support an order-of-magnitude EOL contact-conductance decrease of approximately 40 % to 50 %. This appendix adopts a representative conductance factor

$$f_c(t) = \frac{G(t)}{G_0}, \qquad f_c(0) = 1, \qquad f_c(T_{\text{mission}}) = 0.55. \tag{A.13}$$

A saturating conductance-degradation trend is then written as

$$f_c(t) = 1 - 0.45\frac{1 - e^{-t/2}}{1 - e^{-T_{\text{mission}}/2}}. \tag{A.14}$$

A.3.2. Mapping conductance degradation to temperature scaling

Garzón et al. [24] report temperature responses for a CubeSat thermal model under different global contact-conductance factors. To convert those results into a continuous scaling law, orbit-mean and orbit-amplitude ratios are defined as

$$r_{\text{mean}}(f_c) = \frac{\bar{T}(f_c)}{\bar{T}(1)}, \qquad r_{\text{amp}}(f_c) = \frac{\Delta T(f_c)}{\Delta T(1)}. \tag{A.15}$$

For example, Table A.2 shows the extracted solar-cell values for $\beta = 0^\circ$.

Table A.2: Fitted orbit-mean and orbit-amplitude scaling with global contact-conductance factor f_c, based on Garzón et al. [24].

f_c	T_{min} °C	T_{max} °C	**Mean** °C	**Mean ratio**	**Amplitude** °C	**Amp. ratio**
1.0	26.39	57.81	42.10	1.000	31.42	1.000
0.2	20.84	100.01	60.42	1.435	79.17	2.520
0.1	17.09	133.90	75.49	1.793	116.81	3.718

The extracted ratios are fitted with a power-law form,

$$r(f_c) \approx f_c^b, \tag{A.16}$$

where the exponent b acts as a sensitivity index. The fitted scaling relationships are shown in Figure A.3, and the resulting exponents are summarized in Table A.3.

For a beta-dependent exponent, linear interpolation is used:

$$b(\beta) = b(0^\circ) + \frac{\beta}{30^\circ}\left[b(30^\circ) - b(0^\circ)\right], \qquad 0^\circ \le \beta \le 30^\circ. \tag{A.17}$$

At fixed degradation level f_c, the corresponding mean and amplitude ratios are

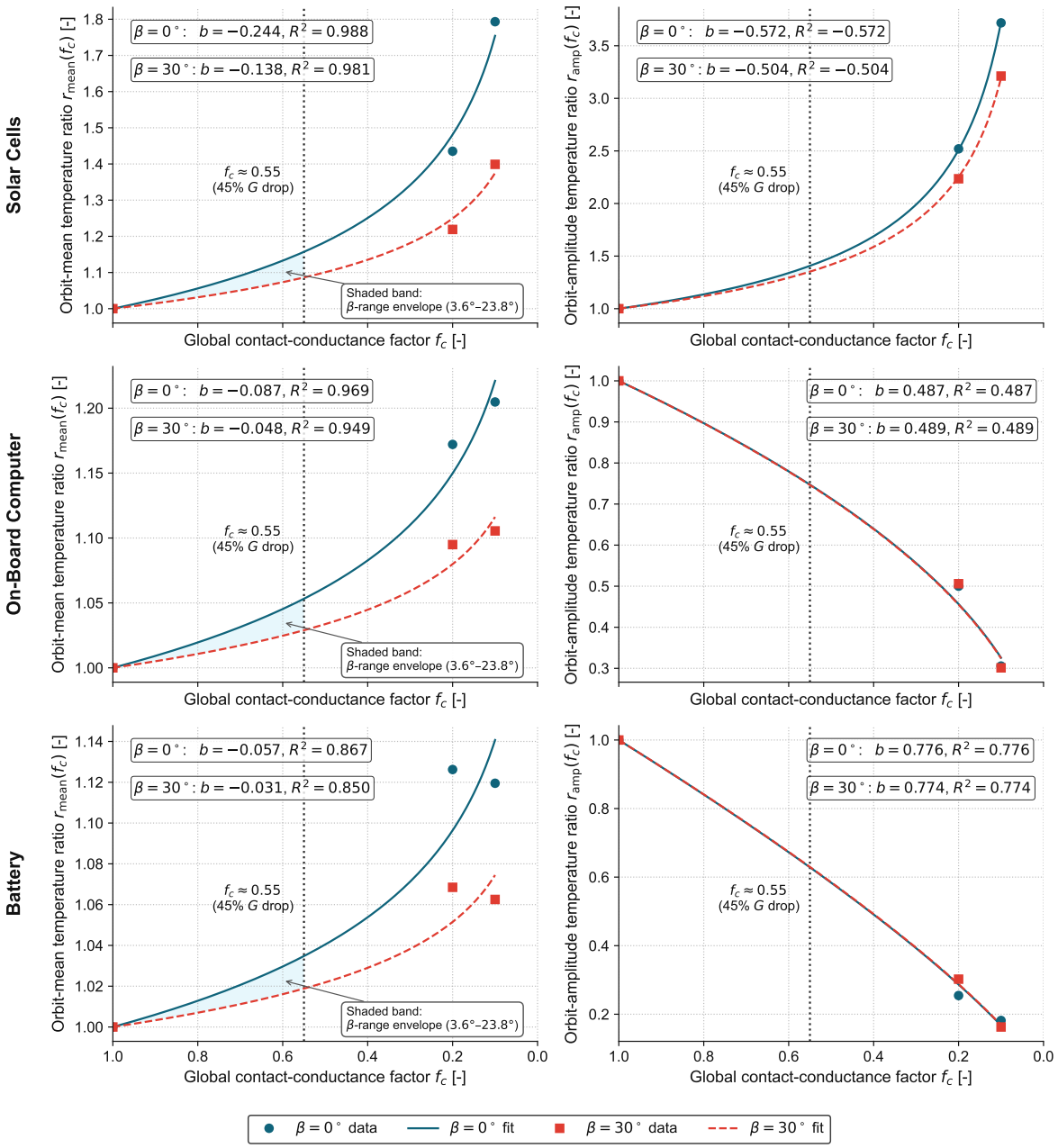


Figure A.3: Fitted orbit-mean and orbit-amplitude temperature scaling with global contact-conductance factor f_c, based on Garzón et al. [24].

Table A.3: Fitted power-law exponents for orbit-mean and orbit-amplitude temperature scaling with contact-conductance factor f_c.

β	**Component**	b_{mean}	$r_{\text{mean}}(f_c)$	b_{amp}	$r_{\text{amp}}(f_c)$
0°	Solar cells	-0.244	$f_c^{-0.24}$	-0.572	$f_c^{-0.57}$
30°	Solar cells	-0.138	$f_c^{-0.14}$	-0.504	$f_c^{-0.50}$
0°	On-Board Computer (OBC)	-0.087	$f_c^{-0.09}$	+0.487	$f_c^{+0.49}$
30°	On-Board Computer (OBC)	-0.048	$f_c^{-0.05}$	+0.489	$f_c^{+0.49}$
0°	Battery	-0.057	$f_c^{-0.06}$	+0.776	$f_c^{+0.78}$
30°	Battery	-0.031	$f_c^{-0.03}$	+0.774	$f_c^{+0.77}$

$$r_{\text{mean}}(f_c, \beta) = f_c^{b_{\text{mean}}(\beta)}, \qquad r_{\text{amp}}(f_c, \beta) = f_c^{b_{\text{amp}}(\beta)}. \tag{A.18}$$

At $f_c = 0.55$, representative EOL scaling factors are summarized in Table A.4.

Table A.4: Component-wise orbit-mean and orbit-amplitude scaling at $f_c = 0.55$, relative to the nominal $f_c = 1$ case.

Component	r_{mean} **range [-]**	$\Delta\bar{T}$ **vs. nominal [%]**	r_{amp} **range [-]**	ΔA **vs. nominal [%]**
Solar cells	1.10 to 1.15	+10 to 15	1.36 to 1.40	+36 to 40
MCU	1.03 to 1.05	+3 to 5	≈ 0.75	≈ -25
Battery	1.02 to 1.03	+2 to 3	≈ 0.63	≈ -37

These results show why conductance degradation is better treated as a predictor-validity problem than as a bounded labelled event. It can change orbit-mean temperatures, orbit amplitudes, and inter-node coupling over long timescales. In this thesis, these effects motivate predictor-mismatch and temperature-bias robustness scenarios rather than labelled COND injections.

A.4. Implications for Predictor-Mismatch Scenarios

The deferred fault families support the robustness tests used later in the thesis. SPIN motivates sensitivity to panel-amplitude, phase, and ripple mismatch. DEG motivates slow positive thermal bias, especially on exposed panels. COND motivates node-specific offsets and changes in orbit-envelope amplitude or thermal coupling.

These effects are not represented as labelled train/validation/evaluation events because they do not have clean, short onset-offset semantics in the simplified benchmark. However, they are operationally important for a residual-based detector because they can make a fixed expected-temperature predictor increasingly misaligned with observed telemetry. This is why predictor-mismatch and temperature-bias scenarios are included in the evaluation.

B

Illustrative Scoring-Family Mechanisms

This appendix provides the illustrative mechanism examples used to support the scoring-family screening in Chapter 8. The examples are not intended as final candidate implementations. Their purpose is to show how each scoring family produces anomaly evidence and what assumptions are implicit in the score.

All examples use the same injected phase-shift anomaly. A short segment of temperature telemetry is shifted forward in phase near a sunlight-to-eclipse transition, producing a local mismatch between observed and nominal behaviour. Two evaluation windows are used: Window A near the transition onset, and Window B near the centre of the shifted region. This common example allows the scoring mechanisms to be compared visually.

B.1. Common Working Example

The working example is a local phase-shift injection in a nominal temperature orbit. Within the injected region, the observed telemetry samples values from a future portion of the nominal curve, so the waveform appears earlier than expected. This is not used as a final benchmark fault class; it is a compact example for visualizing scoring behaviour across families.

This type of phase-shift anomaly is consistent with timing/phase misalignment mechanisms, such as a shift in the effective thermal phase relative to orbit (e.g., mode/timeline changes, scheduling or clock offsets, or delayed/advanced actuation relative to eclipse entry). The injected anomaly, as well as the evaluation windows can be seen in Figure B.1.

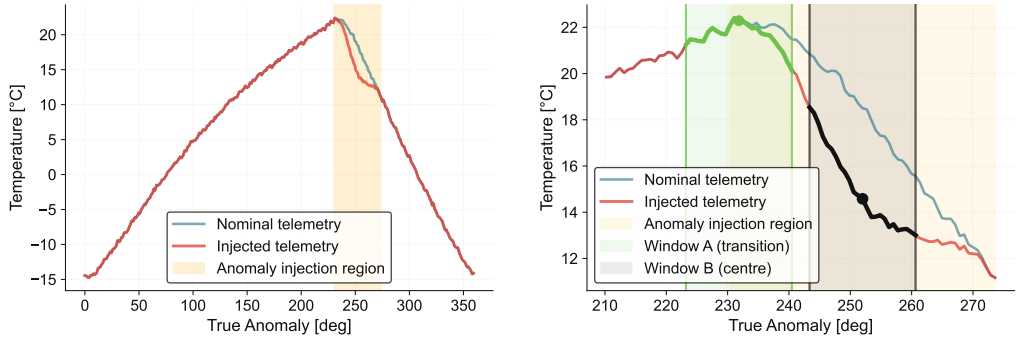


Figure B.1: Injected phase-shift anomaly used as the common working example, with Window A at transition onset and Window B near the injection centre.

B.2. F1 | Rules, Limits, Simple Statistics

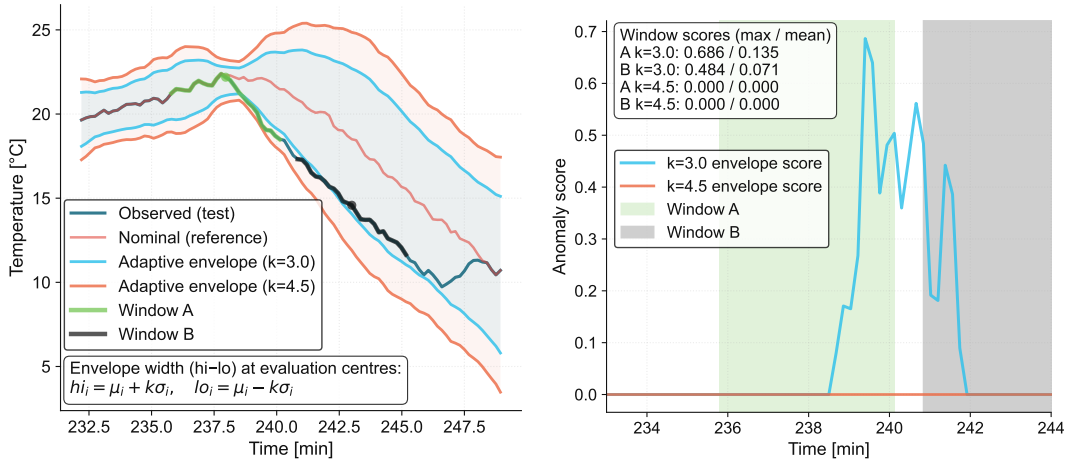
To help visualize how the F1 family (Rules, Limits, Simple Statistics) approaches anomaly scoring, we use *adaptive limits*. This provides a simple statistical baseline that estimates a local centre line and local variability, then scores only an exceedance beyond the resulting envelope. We implement adaptive limits using a moving z-score threshold [9]. Assuming the nominal signal is locally approximately stationary, we estimate a local mean μ_i and standard deviation σ_i (via a symmetric rolling window) and define lower l_i and upper u_i sample varying bounds:

$$l_i = \mu_i - k\sigma_i, \quad u_i = \mu_i + k\sigma_i \tag{B.1}$$

As can be seen in Figure B.2a, we use $k = 3$ (a common "3 sigma" choice) and also evaluate $k = 4.5$ to illustrate the effect of a wider envelope. The anomaly score is the 'magnitude of violation', which is zero inside the envelope, and increases with distance beyond the bounds:

$$s_i = \max(0,\ l_i - T_i^*) + \max(0,\ T_i^* - u_i) \tag{B.2}$$

where T_i^* indicates the temperature at the sample test point, and s_i is the corresponding anomaly score at that sample. Aggregating s_i over a window using max or mean corresponds to peak severity versus average severity. In Figure B.2b, the narrower $k = 3$ envelope produces nonzero exceedances (particularly in Window A), whereas the wider $k = 4.5$ envelope eliminates these exceedances, demonstrating the sensitivity trade off controlled by k.



(a) Moving mean and standard-deviation envelopes over the injected region.

(b) Envelope-violation scores for two moving z-score thresholds.

Figure B.2: F1 adaptive-envelope scoring under different moving z-score thresholds.

B.3. F2 | Forecasting/Prediction Residual

To help visualize how the F2 family (Forecasting/Prediction Residual) approaches anomaly scoring, we implement a simple short-horizon predictor trained on nominal telemetry, and score anomalies using the resulting difference between the observed and predicted values (residual). This is a common technique used in time-series Anomaly Detection (AD) [9], where unexpected behaviour is indicated by unusually large prediction errors relative to what the nominal model can explain.

In this illustrative implementation, the predictor exploits the strong orbit-driven periodicity of temperature telemetry. We compute an orbit phase $\phi_i \in [0, 1)$ for each sample and fit a *phase-binned conditional mean* model on nominal data. Specifically, we discretize the orbit into B bins and compute, for each bin b, the average nominal temperature of all samples whose phase falls in that bin $\bar{T}_b$:

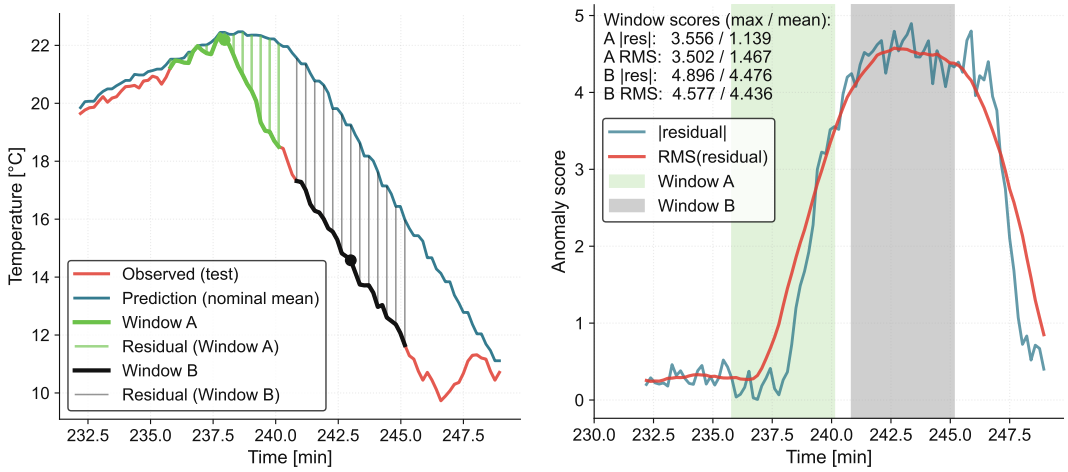
$$\bar{T}_b = \text{mean}(T_i \mid \phi_i \in \text{bin } b) \tag{B.3}$$

At run time, the one-step prediction is the mean corresponding to the current phase bin, $\hat{T}_i = \bar{T}_{b(\phi_i)}$. Given an observed test sample T_i^*, the residual is $r_i = T_i^* - \hat{T}_i$, and the anomaly score is the residual

magnitude $s_i = |r_i|$. To reduce sensitivity to point noise and emphasize sustained mismatch, we also compute a smoothed score using a moving root mean square of the residual over a short window of samples n:

$$s_i^{\text{RMS}} = \sqrt{\frac{1}{|n|}\sum_{j=1}^{n} r_j^2} \tag{B.4}$$

Figure B.3 shows both the prediction mechanism and the resulting scores. In Figure B.3a, the nominal predictor $\hat{T}_i$ is overlaid on the observed test stream T_i^*, and vertical connectors visualize the residual r_i within two evaluation windows. In Figure B.3b, the corresponding score traces are shown, with shaded regions indicating Window A and Window B. Window level summaries (max and mean) provide peak severity and average severity respectively. In this instance, Window B exhibits a consistently larger residual magnitude (for example, mean $|r|$ and mean RMS are both higher in Window B than Window A), reflecting a stronger and more sustained mismatch between the observed telemetry and the nominal predictor.



(a) Observed temperature, baseline prediction, and evaluation windows.

(b) Absolute and smoothed RMS residual anomaly scores.

Figure B.3: F2 residual-based scoring using an orbit phase-conditioned baseline predictor.

B.4. F3 | Reconstruction

To help visualize how the F3 family (Reconstruction) approaches anomaly scoring, we use Principal Component Analysis (PCA) as a nominal representation and score a test window by how well it can be reconstructed from the nominal subspace. This is a common approach in multivariate projection methods for monitoring: if the nominal model captures the typical structure of the data, then abnormal behaviour appears as a large reconstruction residual. This residual is often quantified by the squared prediction error (SPE) or the Q statistic [36]. This framing is widely used in process monitoring and diagnosis with projection-based models [36], while the underlying PCA mechanics are summarized clearly in [63].

We first form sliding windows of length w (here $w = 25$) and treat each window as a vector in $\mathbb{R}^w$. Then, N nominal windows are stacked into a matrix, yielding $X_{\text{nom}} \in \mathbb{R}^{N\times w}$. By averaging the windows across one another we obtain an "average nominal window shape" $\mu \in \mathbb{R}^w$, representing the typical temperature pattern over the window indices:

$$\mu = [\mu_1, ..., \mu_{25}] = \frac{1}{N}\sum_{n=1}^{N} X_{\text{nom}}[n, :] \tag{B.5}$$

Subtracting μ from each nominal window centres the data, producing the matrix $X_c = X_{\text{nom}} - \mu$ that represents the variation around the average window shape. This centring step is essential to PCA since it is designed to capture variance about the mean, rather than being dominated by absolute offset.

From here, PCA aims to describe how each of these vectors varies around the average window by a reduced set of features. In principle, PCA assumes the features, or directions, with the largest variances across the signals are the most “important”, or most principal [63]. It aims to find a set of principal directions in $\mathbb{R}^w$ that capture the dominant patterns of variation in the centred nominal windows such that: PC1 captures the *largest variance* across the nominal windows, PC2 captures the next largest variance, subject to being orthogonal to PC1, and so on.

A standard way to compute these directions is by singular value decomposition (SVD), which produces the following decomposition to represent the nominal window vectors (centred around the mean) [63]:

$$X_c = U\Sigma V^T \tag{B.6}$$

The rows of V^T provide the principal directions. Each direction has an associated “importance” given by the singular values in Σ (equivalently eigenvalues of the covariance). A larger singular value indicates that the corresponding direction explains more variance. In our setting, each principal direction can be interpreted as a template waveform describing one dominant way temperature snippets vary around the nominal mean. For example, the leading modes often resemble variations in:

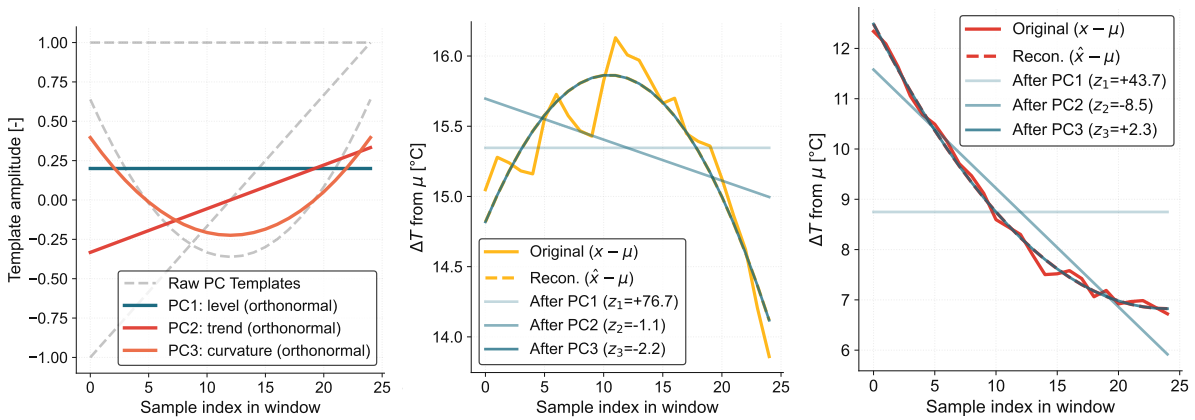
- Overall level (offset / mean temperature): a direction that is roughly flat across the window
- Local trend (heating or cooling slope): a direction that looks like an increasing/decreasing ramp
- Curvature or transition sharpness: a direction that bulges up or down (concave/convex)

These examples are interpretive rather than guaranteed. PCA learns whichever patterns explain the most variation in the nominal windows.

To obtain a compact nominal representation, we retain only the top k principal directions (here $k = 3$), defining a low-dimensional nominal subspace. The key computational benefit is that the principal directions are taken to be orthonormal. This allows any centred window to be (approximately) expressed as a weighted sum of these directions, and the weighted are objected by simple projections (dot products):

$$x_c \approx z_1 v_1 + z_2 v_2 + \cdots + z_k v_k \tag{B.7}$$

Furthermore, this assumption motivates the selection of directions that are unique and non-redundant in comparison to one another. Using the interpretive examples of overall level, local trend, and curvature or transition sharpness, Figure B.4 visualizes how a signal is reconstructed using principal components.



(a) Illustrative shapes of the first three principal components (level, trend, curvature).

(b) Example reconstruction of Window A after linear combination of the principal components with corresponding weights $z = \{z_1, z_2, z_3\}$.

(c) Example reconstruction of Window B after linear combination of the principal components with corresponding weights $z = \{z_1, z_2, z_3\}$.

Figure B.4: PCA reconstruction mechanism using principal components and progressive window reconstruction.

Projection and Reconstruction of a Test Window:

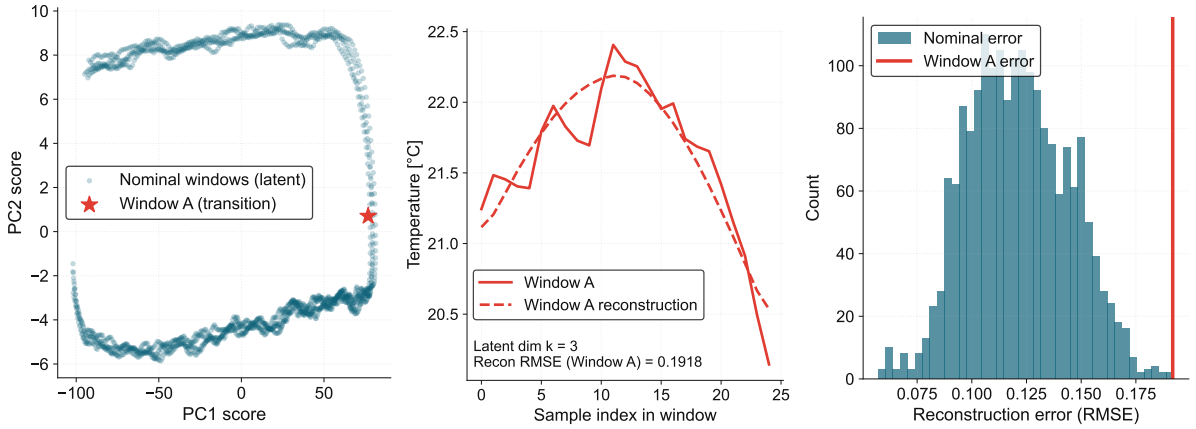
When considering a test window $x^* \in \mathbb{R}^w$ for AD, we work through the following steps:

1. **Projection**: converts the 25-D window into k scalars that quantify how much of each learned principal component is present in the test window.
   In principle, we centre the window using the nominal mean and project it onto the retained subspace to obtain coefficients:
   $$z_j = x_c^* \cdot v_j, \quad z = x_c^* V_k^T \tag{B.8}$$
2. **Reconstruction**: rebuilds the best approximation $\hat{x}^*$ using only the retained directions:
   $$\hat{x}_c^* = z^* V_k, \quad \hat{x}^* = \mu + \hat{x}_c^* \tag{B.9}$$

The anomaly score is then derived from the reconstruction residual, using SPE/Q-statistic style reasoning [36]. In our implementation we report per-window RMSE between x^* and $\hat{x}^*$:

$$s(x^*) = \sqrt{\frac{1}{w}\sum_{j=1}^{w}\left(x_j^* - \hat{x}_j^*\right)^2} \tag{B.10}$$

Figure B.5 and Figure B.6 illustrate this mechanism for the two evaluation windows. The left panel (Figure B.5a/Figure B.6a) shows the nominal windows in the learned latent space of the two principal components, and the location of the evaluated test window. This yields the coefficients $z \in \mathbb{R}^k$. The middle panel (Figure B.5b/Figure B.6b) overlays the original test windows and its reconstruction as the weighted sum of the k principal components from the nominal PCA subspace. The right panel (Figure B.5c/Figure B.6c) places the test window's reconstruction error against the distribution of nominal reconstruction errors.



(a) Window A in the learned two-component latent space.

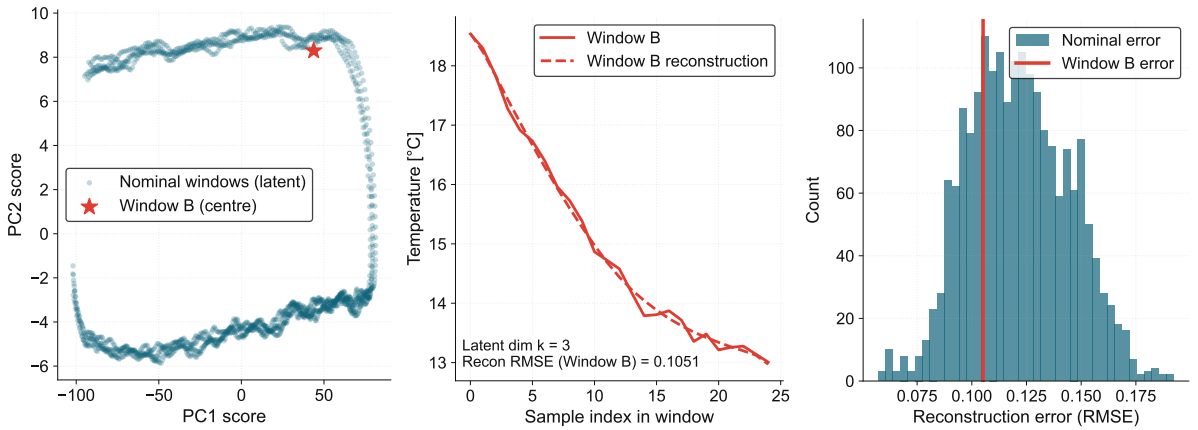
(b) Pointwise reconstruction of Window A using three retained components.

(c) Window-level reconstruction error relative to nominal windows.

Figure B.5: F3 PCA reconstruction scoring for Window A (transition window)

B.5. Window Feature Space for F4-F6

Before illustrating how the F4 to F6 families operate, we first describe the window feature space used in the examples. Like F3, these families evaluate short windows of telemetry. However, assessing each raw window $w \in \mathbb{R}^w$ directly can be costly and statistically challenging in practice. For F4, this requires computing distances from a test window to many stored $\mathbb{R}^w$ nominal windows, increasing memory and compute. For F5, fitting a probability model in w dimensions quickly becomes difficult to estimate robustly as dimensionality grows. For F6, learning a normal region in raw window space is possible, but the complexity of training and evaluation typically increases with dimension.



(a) Window B in the learned two-component latent space. **(b)** Pointwise reconstruction of Window B using three retained components. **(c)** Window-level reconstruction error relative to nominal windows.

Figure B.6: F3 PCA reconstruction scoring for Window B (centre window)

A common approach is therefore to map each window into a lower-dimensional feature vector $f(x) \in \mathbb{R}^d$, designed to capture the prominent window geometry while reducing computation and memory storage. These features can be self-defined (for example, mean, slope, amplitude, curvature) or learned (for example, PCA directions). In this thesis, we choose a two dimensional feature space to keep the mechanism figures interpretable in 2D. Specifically, for each sliding window of width $w = 25$, we compute the window mean and the window slope (the slope of a least squares linear fit across the 25 samples). This produces one feature point per window.

The resulting feature space for a single nominal orbit is shown in Figure B.7. The left panel highlights an example window and its derived mean and slope, while the right panel shows how windows trace a structured curve in $(\text{mean}, \text{slope})$ space as orbit phase progresses. In practice, F4 to F6 operate on feature points collected over many orbits, which produces a thicker band rather than a single curve. The choice of feature set is important: it directly determines what variations are considered “near” or “far” (F4), “likely” or “unlikely” (F5), and “inside” or “outside” the normal region (F6).

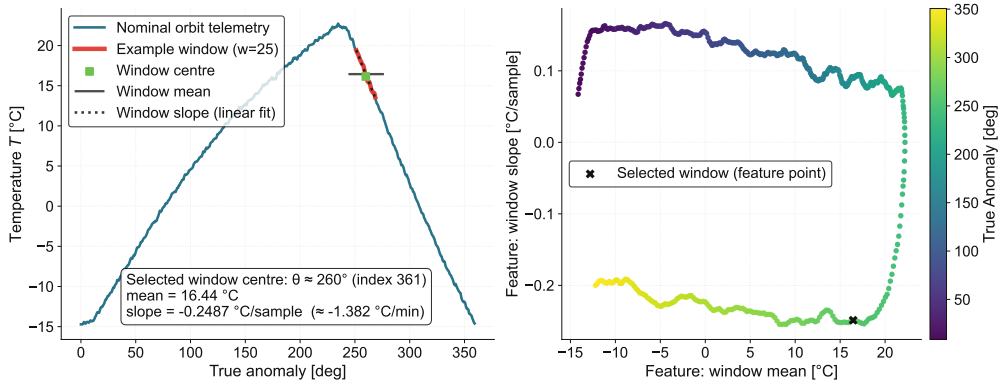


Figure B.7: Mapping orbit windows into a low-dimensional feature space (window mean vs. window slope) using sliding windows, for F4-F6 anomaly scoring families.

B.6. F4 | Proximity / Distance

To help visualize how the F4 family (Proximity / Distance) approaches anomaly scoring, we implement a k-nearest neighbour (kNN) score based on local neighbourhood density. The intuition is that nominal windows cluster in feature space, whereas anomalous windows fall in sparse regions and therefore lie farther from their nearest nominal neighbours [9]. This approach is ubiquitous not only in AD, but also in many Machine Learning (ML) applications.

Each raw window $x \in \mathbb{R}^w$ is mapped to a low-dimensional feature point $f = (\mu, m)$, where μ is the window mean and m is the window slope. For a test window $x^* \in \mathbb{R}^w \Rightarrow f^*$, we compute Euclidean distances to all nominal feature points $\{f_i\}$:

$$d(f^*, f_i) = \sqrt{(\mu^* - \mu_i)^2 + (m^* - m_i)^2} \tag{B.11}$$

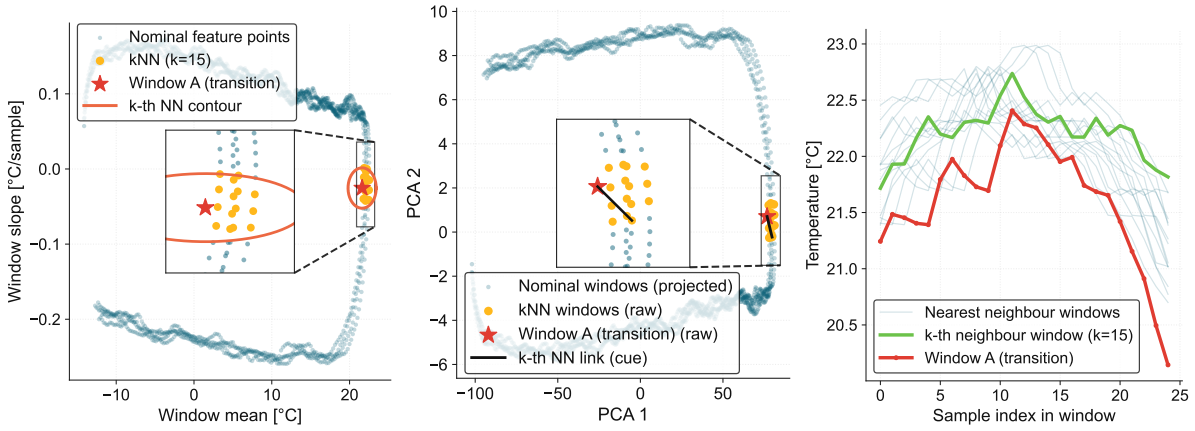
Sorting these distances yields $d_{(1)} \leq \cdots \leq d_{(k)}$. The anomaly score is taken as the distance to the k-th nearest neighbour, $s(f^*) = d_{(k)}$, which corresponds to the radius of the smallest circle centred at f^* that contains k nominal points. In the left panel (Figure B.8a, Figure B.9a), the orange contour visualizes this equal-distance boundary (it appears elliptical due to feature scaling).

While kNN can also be applied directly in raw window space $\mathbb{R}^w$, using

$$d(x^*, x) = ||x^* - x|| = \sqrt{\sum_{j=1}^{w} (x_j^* - x_j)^2} \tag{B.12}$$

distances in high dimensions are hard to visualize. We therefore project windows into a 2D PCA view purely for interpretation: the middle panel (Figure B.8b, Figure B.9b) shows the test window and the selected neighbour windows in the projected space. The score itself remains the distance-based kNN criterion. Here we again use PCA to determine the k^{th} nearest neighbour by its two principal components, as seen in the middle panel. The right panel (Figure B.8c, Figure B.9c) then overlays the evaluated test window against its k-th neighbour window, illustrating how the distance score corresponds to a perceptible difference in window shape.

Here we use $k = 15$ purely for illustrative purposes. In practice, a smaller k yields a more sensitive but potentially noisy AD, whereas a larger k yields smoother behaviour, but may miss anomalies.



(a) Feature space showing Window A and its k nearest nominal neighbours.

(b) PCA projection of raw windows highlighting Window A versus retrieved neighbours.

(c) Time-domain comparison of Window A with its k-th nearest nominal window

Figure B.8: F4 kNN proximity scoring for Window A in mean–slope feature space.

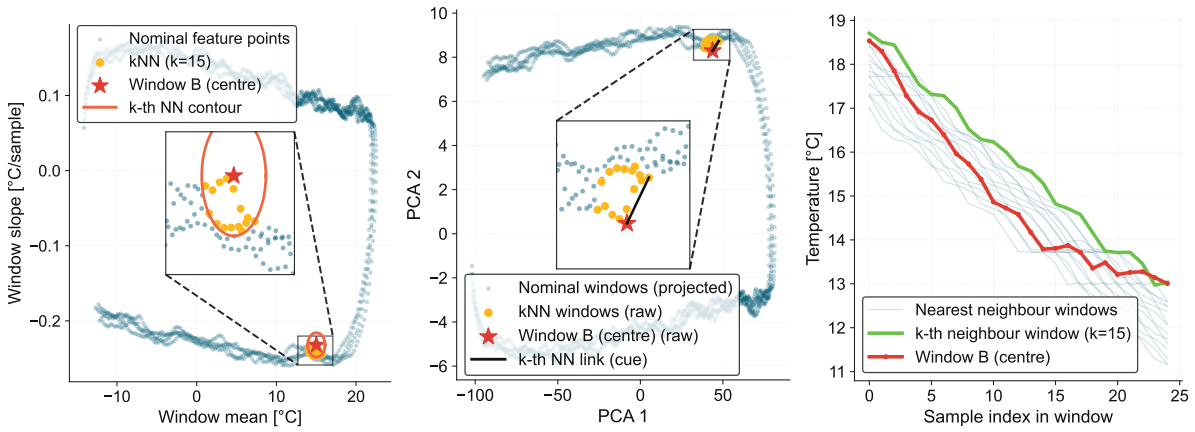
B.7. F5 | Distribution / Density

To help visualize how the F5 family (Distribution / Density) approaches anomaly scoring, we fit a multivariate Gaussian to the nominal feature points. The intuition is that nominal windows occur in high-density regions of a stochastic model, whereas anomalous windows lie in low-density regions [9].

Let each window be mapped to a feature vector $f = (\mu, m) \in \mathbb{R}^d$ (here $d = 2$). A multivariate normal density is defined by:

$$f(f) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{1}{2}(f - \mu)^T \Sigma^{-1} (f - \mu)\right) \tag{B.13}$$

where μ is the mean (centre) and Σ is the covariance (spread and correlation). Fitting this model to the nominal feature points yields the Gaussian mean shown in Figure B.10a.



(a) Feature space showing Window B and its k nearest nominal neighbours.

(b) PCA projection of raw windows highlighting Window B versus retrieved neighbours.

(c) Time-domain comparison of Window B with its k-th nearest nominal window

Figure B.9: F4 kNN proximity scoring for Window B in mean–slope feature space.

A convenient score for "how improbable" a point is under this model is the Mahalanobis squared distance [9], which appears in the exponent of the Gaussian:

$$D^2(f^*) = (f^* - \mu)^T \Sigma^{-1} (f^* - \mu) \tag{B.14}$$

In 2D, sets of constant D^2 form elliptical equal-density contours, as shown by the equal-M^2 curves in Figure B.10a. Equivalently, we can score by the negative log-likelihood (NLL):

$$\text{NLL}(f) = \frac{1}{2} \left(D^2(f) + \log |\Sigma| + d \log (2\pi) \right) \tag{B.15}$$

For a fixed trained model, $\log |\Sigma|$ and $d \log(2\pi)$ are constants, so ranking points by NLL is equivalent to ranking them by D^2: larger D^2 implies lower density and a more anomalous score. This is reflected in Figure B.10b, where the evaluated windows have higher NLL than typical nominal points.

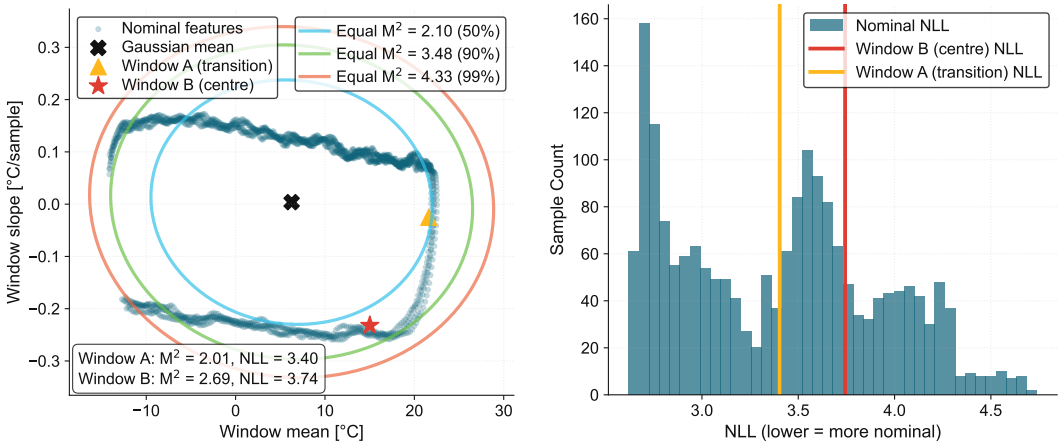
Of note, the nominal mean and slope features here do not follow a single Gaussian distribution (the nominal points form a curved, multi-regime structure). We therefore use the Gaussian primarily as an illustrative example of the F5 mechanism: the score depends on the density of nominal sampling, so regions with sparse nominal coverage are naturally treated as less probable (more anomalous). In practice, this motivates using richer density models that can represent multi-modal or non-Gaussian nominal structure (e.g., Gaussian mixture models, kernel density estimation, or other flexible generative models), while preserving the same core idea: low nominal density implies higher anomaly score.

B.8. F6 | Boundary and Ensembles

Boundary

Boundary methods learn a "region of normality" from nominal data, and then score a test instance by whether it lies outside that region. This is conceptually similar to F1 (inside = normal, outside = anomaly), but the boundary is learned in a window feature space rather than defined as point-wise limits on the raw telemetry.

To help visualize how this subfamily approaches anomaly scoring, we implement a Gaussian ellipse boundary in the $(\text{mean}, \text{slope})$ feature space. Similarly to F5, we fit a Gaussian model (μ, Σ) to the nominal feature points f, and compute the Mahalanobis squared distance $D^2(f)$ and corresponding distance $d(f) = \sqrt{D^2(f)}$. The boundary is defined by a threshold d_{thr}, chosen as the empirical p-th percentile of nominal distances (here $p = 99\%$). This produces an elliptical decision boundary $D^2(f) =$ const, as seen in Figure B.11a.



(a) Gaussian nominal model with equal Mahalanobis-distance contours and evaluation windows.

(b) Negative log-likelihood distribution for nominal windows with the test window scores highlighted.

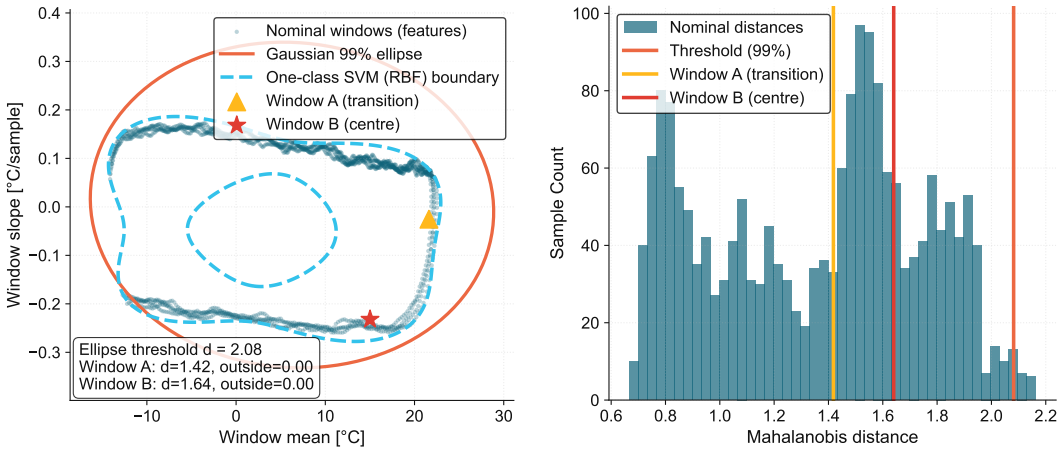
Figure B.10: F5 density-based scoring under a multivariate Gaussian model in mean–slope feature space.

Analogous to the “magnitude of violation” used in F1, the anomaly score is zero inside the region and increases with distance beyond the boundary:

$$s(f) = \max\left(0,\, d(f) - d_{\text{thr}}\right) \tag{B.16}$$

As seen in Figure B.11b, both evaluation windows lie within the 99% ellipse (outside score = 0), since $d_{\text{thr}} = 2.08$, while $d_A = 1.42$ and $d_B = 1.64$.

Since the nominal features are not well approximated by a single Gaussian ellipse, we also show a one-class SVM with an RBF kernel, which can learn a non-elliptical decision boundary around the support of the nominal data. Conceptually, the one-class SVM learns a function $f(f)$ whose zero level set $f(f) = 0$ acts as the boundary: points with $f(f) < 0$ are treated as outliers (novelty detection) [62]. A closely related formulation is SVDD (support vector data description), which similarly encloses nominal data with a flexible kernelized boundary [67].



(a) One-class Gaussian and SVM boundaries relative to the evaluation windows.

(b) Mahalanobis-distance distribution with evaluation window scores highlighted.

Figure B.11: F6 boundary-based scoring: outlierness defined by distance outside a learned normal region.

Ensembles

To help visualize how the Ensembles subfamily of F6 approaches anomaly scoring, we implement a simple Isolation Forest, which scores an instance based on how quickly it can be isolated by random partitioning. An isolation tree recursively partitions the feature space using random splits, first selecting a feature and then choosing a split uniformly between that feature's minimum and maximum values within the current node. This process is visualized in Figure B.12. The intuition is that points in dense nominal regions tend to survive many random splits before being isolated, yielding longer path lengths, whereas points in sparse regions tend to get separated more quickly, yielding shorter path lengths.

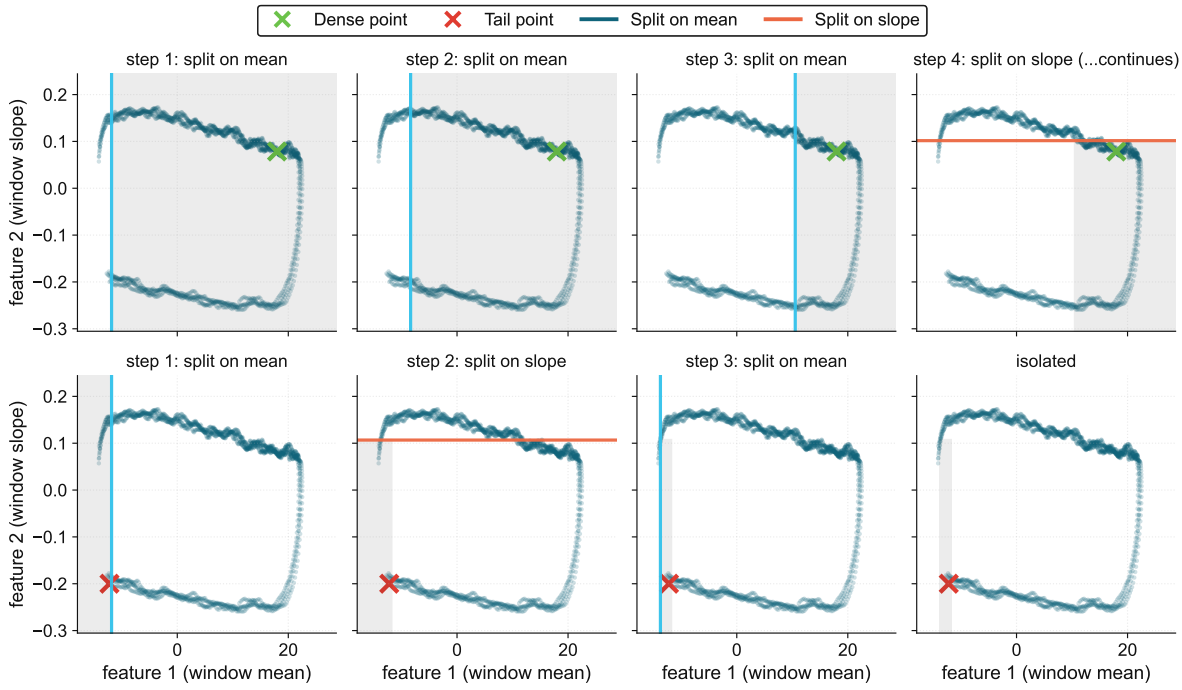


Figure B.12: Isolation Forest partitioning mechanism using random axis-aligned splits.

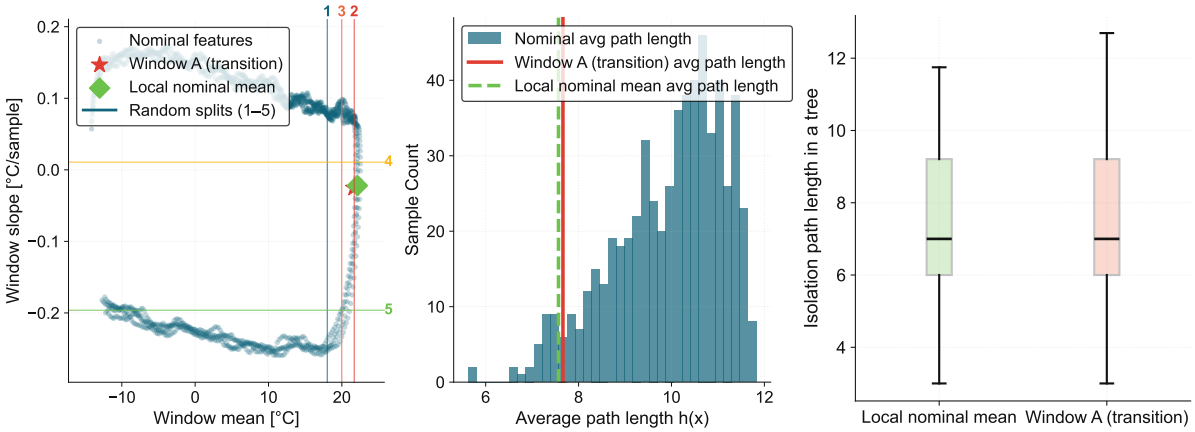
Isolation Forest aggregates this behaviour over many trees, each trained on a random subsample of the nominal data. In our implementation, we build 80 trees, each using a subsample of size $\psi = 256$. Splitting stops when either (i) a maximum tree depth has been reached, (ii) the node contains one (or zero) points, or (iii) all points have the same value along that feature, meaning no further split is possible.

For a test feature vector f, we compute its average path length across the ensemble $h(f)$, and convert this to an anomaly score using the standard Isolation Forest mapping [39]:

$$s(f) = 2^{-\frac{h(f)}{c(\psi)}} \tag{B.17}$$

where $c(\psi)$ is a normalizing constant equal to the expected path length of an unsuccessful search in a binary search tree of size ψ. Essentially, instances with shorter average path lengths yield larger anomaly scores.

To aid interpretation, we also plot a local nominal reference, defined as the mean of the k closest nominal feature points (using Mahalanobis-squared distance), shown alongside the test window in feature space and in the path-length summaries. This approach for the Ensembles subfamily of F6 is given in Figure B.13 and Figure B.14b for Window A (transition) and Window B (centre), respectively.

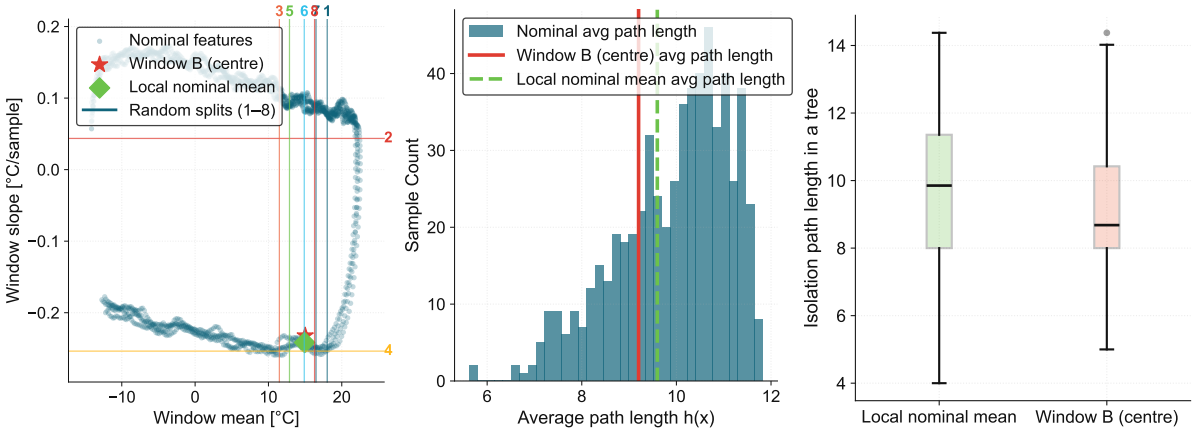


(a) Feature space with Window A and a local nominal reference point. Random splits indicate the mechanism behind the isolation of the test point.

(b) Distribution of average path lengths over nominal windows with Window A highlighted (shorter = more anomalous).

(c) Path-length comparison for Window A versus the local nominal reference, averaged over multiple trees.

Figure B.13: Isolation Forest scoring for Window A: path length as an outlierness measure.



(a) Feature space with Window B and a local nominal reference point. Random splits indicate the mechanism behind the isolation of the test point.

(b) Distribution of average path lengths over nominal windows with Window B highlighted (shorter = more anomalous).

(c) Path-length comparison for Window B versus the local nominal reference, averaged over multiple trees.

Figure B.14: Isolation Forest scoring for Window B: path length as an outlierness measure.

C

Screening Rationale for Anomaly Scoring Family Ratings

This appendix provides the rationale behind the **+**/**0**/**–** ratings in the screening matrices. Ratings should be interpreted as *relative* suitability under Delfi Twin's telemetry and application constraints. Where applicable, rationales draw on reported method assumptions and limitations in prior surveys and spacecraft-oriented benchmarking guidance (e.g., [9, 6, 21, 35]).

C.1. Telemetry-Driven and Data-Dependent Screening (Q1 to Q8)

Table C.1: Q1 rationale: handling variable sequence coverage and duration (gaps, unequal length, irregular sampling).

Family	Rating	Rationale (key points)
F1 | RLS	**+**	Per-sample checks remain defined under missing segments. No fixed-length requirement; avoids distance/likelihood scaling with length.
F2 | FPR	**0**	Requires sufficient context for forecasting; gaps often require explicit handling (resetting state, masking, interpolation), but manageable with careful implementation.
F3 | R	**0**	Often relies on consistent representations (fixed windows or feature sets). Missingness can be handled (masking, robust losses), but this is part of model design.
F4 | PD	**–**	Distance becomes ill-defined or dominated by unequal coverage/missingness; irregular sampling often requires resampling/alignment that can introduce artifacts.
F5 | DD	**0**	Likelihood typically scales with representation length and missingness; workable if normalized appropriately or if the model explicitly supports missing data.
F6 | BE	**0**	Usually expects fixed feature vectors; tolerates gaps only insofar as features and preprocessing remain well-defined under missingness.

Table C.2: Q2 rationale: handling misalignment of comparable data or key features (phase shifts, dephasing).

Family	Rating	Rationale (key points)
F1 | RLS	**+**	Can be made phase-tolerant via regime-conditioned envelopes or robust summaries; only fragile when rules are tied to fixed indices.
F2 | FPR	**0**	Can tolerate some misalignment if the predictor learns it, but residuals typically inflate for nominal-but-shifted behaviour unless explicitly handled.
F3 | R	**0**	If inputs are aligned window vectors, phase shifts change the vector substantially; sensitivity depends on training (aligned vs shift-invariant features).
F4 | PD	**–**	Misalignment directly corrupts distances under fixed indexing. Elastic distances can mitigate this but add computational and modelling complexity.
F5 | DD	**–**	Phase shifts appear "unlikely" under a fixed-index likelihood model; can be mitigated via regime/state models or shift-invariant features, but increases complexity.
F6 | BE	**0**	Depends strongly on representation: shift-invariant features can help; raw aligned windows generally remain sensitive.

Table C.3: Q3 rationale: handling unknown and variable anomaly length.

Family	Rating	Rationale (key points)
F1 | RLS	**+**	Limits/rules can be evaluated at any time scale; duration is handled downstream in decision logic (persistence/event formation).
F2 | FPR	**+**	Residuals are naturally per time step; can be aggregated over variable windows/horizons without assuming a fixed event length.
F3 | R	**+**	Reconstruction error can be computed per-sample or per-window and aggregated; does not require a fixed anomaly duration by design.
F4 | PD	**0**	Scoring depends on the chosen instance definition (sample vs window). Multi-scale variants exist but add complexity and compute.
F5 | DD	**0**	Same dependency as PD: likelihood is defined on a chosen representation; multi-scale likelihoods are possible but not inherent.
F6 | BE	**0**	Scores are defined per instance/window; variable-length episodes require aggregation and persistence logic.

Table C.4: Q4 rationale: supporting multiple nominal regimes (multi-modal normal geometry).

Family	Rating	Rationale (key points)
F1 | RLS	**+**	Regime-conditioned limits are straightforward (orbit phase bins, mode flags, seasonal baselines); avoids forcing a single global "normal".
F2 | FPR	**0**	A single predictor may average across regimes; robust multi-regime performance often requires explicit conditioning (inputs, separate models, mixture/state structure).
F3 | R	**0**	Single reconstructor can learn an "average" regime; multi-regime support often needs conditioning or mixture/subspace variants.
F4 | PD	**+**	Local neighbourhood comparisons naturally accommodate multiple clusters of nominal data (nearest neighbours, multiple prototypes).
F5 | DD	**+**	Mixture and state-space likelihood models (e.g., GMM/HMM) explicitly represent multi-modal normality.
F6 | BE	**0/–**	A single boundary around multiple clusters can include "holes" and be poorly calibrated; works better with regime conditioning or feature/model choices that encode multi-modality.

Table C.5: Q5 rationale: vulnerability to clustered or repeating anomaly-like patterns (“bulk anomalies”).

Family	Rating	Rationale (key points)
F1 | RLS	**0**	Detects bulk patterns only if they violate the envelope; if the pattern remains within limits, rules provide no discrimination.
F2 | FPR	**0**	If repeating anomaly-like patterns contaminate training, predictors can adapt and residuals shrink; constraints on adaptation and careful training selection are needed.
F3 | R	**0**	Recurring patterns can reconstruct well if learned as nominal; robustness depends on training cleanliness and representation choices.
F4 | PD	**–**	Dense anomalous clusters can appear “normal” under neighbour/prototype logic if they are frequent and tightly clustered.
F5 | DD	**–**	Frequent patterns become high-density/high-likelihood and appear nominal; avoiding this typically requires additional semantics or training constraints.
F6 | BE	**0/–**	With clean nominal training, boundaries/ensembles can reject repeating faults; with contamination or frequent fault-like behaviour, learned normal regions can expand to include them.

Table C.6: Q6 rationale: robustness under high-dimensional representations (e.g., orbit segments vectorized to hundreds of samples).

Family	Rating	Rationale (key points)
F1 | RLS	**0**	Can avoid high-dimensionality by scoring low-cost summaries (mean, slope, extrema), but may miss collective anomalies that require richer representations.
F2 | FPR	**+**	Operates per-sample and can accumulate evidence over time without vectorizing long windows, reducing “curse of dimensionality” issues in distance/density.
F3 | R	**+**	Explicitly designed to represent data in low-dimensional subspaces/latents; can mitigate high-dimensionality when separability holds in the representation.
F4 | PD	**–**	Distances become less discriminative as dimension increases; neighbour/clustering methods degrade significantly in high-D settings.
F5 | DD	**–**	Density estimation and likelihood modelling become brittle in high dimensions unless heavily constrained (parametric structure, low latent).
F6 | BE	**0/+**	Tree-based ensembles can be less distance-sensitive than PD/DD, but performance depends on representation and feature quality; boundary methods can still struggle if the normal set is complex.

Table C.7: Q7 rationale: inference compute suitability for real-time on-board use (training assumed offline).

Family	Rating	Rationale (key points)
F1 | RLS	**+**	Very low compute and memory footprint; simple arithmetic on streaming telemetry.
F2 | FPR	**0**	Compute depends on the predictor; small models can be streamable, but model choice controls feasibility.
F3 | R	**0**	Subspace reconstructions can be lightweight; neural reconstructions can be expensive. Inference cost is strongly model-dependent.
F4 | PD	**–**	Nearest-neighbour style methods incur expensive test-time comparisons to many references and can be difficult to bound on-board.
F5 | DD	**0**	Model-dependent: simple parametric likelihoods can be cheap; richer mixture/state models can be expensive and memory heavy.
F6 | BE	**0/+**	Isolation Forest and small ensembles can be compute-friendly; boundary methods vary with feature dimension and model form.

Table C.8: Q8 rationale: dependence on “anomalies are rare” reasoning (vulnerability to non-rare/bulk anomalies).

Family	Rating	Rationale (key points)
F1 | RLS	**+**	Limit violation is semantic deviation from bounds, not explicitly frequency-based.
F2 | FPR	**+**	Residuals measure deviation from predicted dynamics, not rarity; bulk patterns still trigger if they remain unpredictable under the nominal model.
F3 | R	**+**	Reconstruction error measures representational mismatch, not probability mass; bulk patterns remain anomalous if they do not reconstruct under a nominal representation.
F4 | PD	**0**	“Far from neighbours” often correlates with rarity, even if not explicitly frequency-based; clustered anomalies can appear nominal if they form dense neighbourhoods.
F5 | DD	**–**	Low likelihood/low density is fundamentally a rarity notion; frequent fault-like patterns can become high probability and appear nominal.
F6 | BE	**0/–**	Isolation Forest explicitly benefits from “few and different”, while boundary methods can also be biased if frequent fault-like behaviour is learned into the normal region.

C.2. Application-Driven Screening (Q9 to Q11)

Table C.9: Q9 rationale: early fault warning (mean shifts, slow drift, and runaway trends).

Family	Rating	Rationale (key points)
F1 | RLS	**0**	Early warning is possible if scoring uses drift/rate features and regime-conditioned baselines; standard limit exceedance is typically late.
F2 | FPR	**+**	Residuals tend to shift early for mean shifts, drift onset, and runaway behaviour, provided the predictor is not adapting faults away.
F3 | R	**0**	Can detect early if the nominal representation is tight and does not absorb drift; sensitivity depends strongly on representation and training.
F4 | PD	**0**	Can detect early if using drift-sensitive features and appropriate references; performance depends on representation quality and reference coverage.
F5 | DD	**0**	Can detect early if the nominal model is well calibrated and regime-conditioned; small shifts and slow drift can remain “probable” under flexible models.
F6 | BE	**0/+**	Isolation Forest on low-cost window features often flags early; boundary methods vary widely and may require regime conditioning and careful tuning.

Table C.10: Q10 rationale: human interpretability (comprehensibility) of score-level explanations.

Family	Rating	Rationale (key points)
F1 | RLS	**+**	Directly explainable in the measurement domain (within bounds vs outside bounds); aligns with traditional ops limit checking.
F2 | FPR	**+**	Contrastive explanation (“expected vs observed”) is intuitive; residual magnitude maps directly to “how surprising” the observation is.
F3 | R	**0**	“Could not reconstruct” is explainable but more abstract (subspace/latent representation); feature-level blame can be non-trivial when variables correlate.
F4 | PD	**0**	Can be explained as “far from nominal references”, but depends on chosen features/windowing and distance definition.
F5 | DD	**0**	Probability/likelihood explanations can be non-actionable for non-ML users without careful framing and calibration.
F6 | BE	**0/–**	Isolation Forest can offer model-specific explanations (path length, split features) but is still indirect; boundary geometry is typically less intuitive to operators.

Table C.11: Q11 rationale: configuration effort (complexity) required to achieve robust performance.

Family	Rating	Rationale (key points)
F1 | RLS	**+**	Few sensitive choices; thresholds/limits are straightforward to set and validate; low tuning burden.
F2 | FPR	**0**	Complexity depends on predictor choice and thresholding/event logic; fewer modelling choices than subspace/density families when kept lightweight.
F3 | R	**–**	Configuration can be substantial (representation, latent dimension, training regime, calibration); high sensitivity to design choices.
F4 | PD	**0**	Moderate tuning burden (features, window length, distance metric, neighbour count, reference management).
F5 | DD	**–**	Many modelling choices (distribution family, mixtures/states, covariance structure, regularization); calibration is often sensitive.
F6 | BE	**0/–**	Isolation Forest can be low-effort when small and feature-based; boundary methods often require careful kernel/regularization/conditioning choices.

D

Supporting Evaluation Results

This appendix provides supporting predictor-mismatch and embedded replay results for the consolidated evaluation in Chapter 12. The main text reports the headline predictor-validity, lifecycle, timing, memory, packet-size, power, and adaptive-confirmation conclusions. The tables below provide the detailed measurements from which those conclusions were derived.

D.1. Detailed Predictor-Mismatch Results

The main text summarizes predictor-mismatch behaviour using false-alert occupancy, because that metric directly shows whether a detector mode remains operationally usable when the residual baseline is shifted. Table D.1 provides the corresponding detailed metrics: alert-worthy validation recall, false detector events per day, and node-time alert burden.

Node-time alert burden is the summed fraction of monitored node-time spent in an active detector state. Because six temperature nodes are monitored, a value of 4.589 means that, on average, approximately 4.589 nodes were simultaneously in alert. The false-alert occupancy reported in the main text is the same quantity divided by the six monitored nodes and expressed as a percentage.

Table D.1: Detailed predictor-mismatch validation results for fixed and adaptive detector modes.

Mismatch scenario	Fixed	Common-mode	Node-wise	Hybrid
Alert-worthy validation recall				
Baseline	1.00	1.00	1.00	1.00
Global $+0.5\,°C$	1.00	1.00	1.00	1.00
Ageing ramp	1.00	1.00	1.00	1.00
Battery $+0.5\,°C$	1.00	1.00	1.00	1.00
False detector events per day [d^{-1}]				
Baseline	0.016	0.019	0.032	0.034
Global $+0.5\,°C$	92.863	0.027	0.040	0.037
Ageing ramp	92.129	110.056	0.029	0.026
Battery $+0.5\,°C$	16.155	96.836	0.033	0.034
Node-time alert burden [–]				
Baseline	0.000 64	0.000 65	0.001 08	0.001 10
Global $+0.5\,°C$	4.589	0.000 98	0.001 28	0.001 08
Ageing ramp	4.231	2.877	0.000 97	0.000 94
Battery $+0.5\,°C$	0.771	1.832	0.001 08	0.001 10

D.2. Detailed Embedded Replay Results

The main Chapter 12 reports a consolidated embedded feasibility summary. The following tables provide the underlying timing measurements by replay category, fixed/adaptive mode, and runtime context. These details support the conclusion that routine predictor and detector updates remain sub-millisecond, while the slow timing tail is associated with event-packet emission rather than steady-state processing.

Table D.2: Adaptive embedded timing by event-window category.

Category	Windows	Samples	Weighted mean [ms/sample]	Worst sample [ms]	Max cadence use [%]
Benign	206	55 372	0.591	11.838	0.078921
Telemetry quality	30	9988	0.640	12.627	0.084182
Thermal fault	6	3056	0.631	12.797	0.085310
All replayed windows	242	68 416	0.600	12.797	0.085310

Table D.3: Fixed-versus-adaptive embedded runtime comparison.

Metric	Fixed on-board	Adaptive on-board	Difference
Windows	243	242	−1 benign window
Samples	68 752	68 416	−336 samples
Weighted mean time/sample	0.586 ms	0.600 ms	0.014 ms
Worst observed sample	12.777 ms	12.797 ms	0.020 ms
Maximum cadence utilization	0.085 178 %	0.085 310 %	0.000 132 %

Table D.4: Runtime-context timing breakdown for adaptive embedded replay.

Runtime context	Intervals	Median max [μs]	P95 max [μs]	Max [μs]
Nominal/quiet	12 674	580.4	593.0	624.1
Candidate alarm	39	580.1	591.1	593.4
Event active	1144	581.8	594.4	614.0
Gap/reset	3067	547.8	580.3	604.4
Event packet interval	180	11 234.6	12 040.7	12 796.5

D.3. Memory, Packet-Size, and Power Details

The following tables provide the supporting resource measurements for the embedded feasibility summary. Memory results are based on firmware build sections, packet sizes are measured from replay output, and power impact is estimated from duty cycle and assumed active MCU power. The power values are therefore compute-duty estimates, not direct full-board power measurements.

Table D.5: Incremental memory cost of enabling node-wise adaptive correction.

Metric	Without adaptive	With adaptive	Increase
`.text`	55 360 B	56 944 B	$+1584$ B
`.data`	520 B	520 B	0 B
`.bss`	2808 B	2904 B	$+96$ B
Estimated flash	55 880 B	57 464 B	$+1584$ B
Estimated RAM	3328 B	3424 B	$+96$ B

Table D.6: Packet-size summary from T-mode evaluation replay.

Packet type	Mean size [B]	Max size [B]	Purpose
`D` packet	82.6	96	Periodic diagnostic and timing summary
`E` packet	114.4	132	Compact event alert

Table D.7: Estimated compute-duty power impact of embedded Anomaly Detection (AD).

Quantity	Value	Interpretation
Assumed active MCU power	approx. 34 mW to 36 mW	3.3 V rail, STM32L4 run-mode estimate
Mean processing time	0.586 ms per sample	Overall evaluation T-mode result after embedded spike suppression
Worst observed processing time	12.78 ms per sample	Worst observed event-packet path
Mean compute duty at 15 s cadence	0.003 91 %	Incremental Anomaly Detection (AD) compute duty
Worst compute duty at 15 s cadence	0.0852 %	Worst observed sample path
Estimated mean Anomaly Detection (AD) compute power	approx. 0.0014 mW	Duty-cycle estimate only
Estimated worst-path equivalent compute power	approx. 0.031 mW	Duty-cycle estimate only

D.4. Targeted Adaptive Mismatch Replay Details

The main text summarizes the embedded adaptive-correction confirmation. The following results provide the detailed constant-offset and accelerated ageing-ramp measurements. These tests confirm that the node-wise adaptive correction path is functional on the STM32L4 prototype and can correct representative local predictor mismatch during replay.

The results are not intended to establish flight-safe adaptation. They do not cover all nodes, operating modes, telemetry validity states, or fault families. Their role is to support the narrower claim that the embedded adaptive mechanism is resource-feasible and functionally active on hardware.

Table D.8: Embedded constant-offset mismatch confirmation for a Batt $+0.5$ °C local mismatch.

Metric	Fixed on-board predictor	node-wise adaptive predictor
Node	Batt	Batt
Samples	2000	2000
Replay duration	8.33 h	8.33 h
Applied mismatch	$+0.5$ °C	$+0.5$ °C
Detector events	8	1
Event packets	15	2
Final adaptive bias	N/A	0.4958 °C
Median adaptive bias, final 20%	N/A	0.4992 °C
Median residual, first 20%	0.499 °C	0.2281 °C
Median residual, final 20%	0.488 °C	−0.0112 °C
Time to within ± 0.05 °C of target	N/A	1.84 h
Time to within ± 0.01 °C of target	N/A	2.76 h

Table D.9: Embedded node-wise tracking performance under accelerated Batt ageing-ramp mismatch.

Rows	Duration [h]	Ramp slope [°C/h]	Tracking MAE [°C]	Final error [°C]	Within ± 0.05 °C
20 000	83.33	0.0156	0.0123	-0.0196	1.000
10 000	41.66	0.0312	0.0247	-0.0333	1.000
5000	20.82	0.0624	0.0482	-0.0668	0.561
2500	10.33	0.1259	0.0911	-0.1003	0.056

E

Supporting FUNcube Assessment

This appendix provides the full supporting material for the FUNcube-1 real-telemetry assessment in Chapter 13. The main body uses reduced figures and a compact evidence summary so that the FUNcube analysis remains focused on morphology-level and workflow-level interpretation. The appendix preserves the complete four-panel diagnostic plots and the detailed behaviour-by-behaviour relationship between the synthetic benchmark and the FUNcube stress-test evidence.

The material in this appendix should be interpreted with the same claim boundary used in the main FUNcube chapter. FUNcube-1 provides real on-orbit telemetry with contact gaps, contextual thermal variation, telemetry-quality artifacts, and panel ripple structure. It does not provide labelled Delfi Twin fault events, independent root-cause confirmation, or detector-performance ground truth. The figures and table below therefore support qualitative workflow traceability and benchmark-plausibility assessment, not quantitative flight validation.

E.1. Detailed Synthetic Benchmark/FUNcube Evidence Relationship

The main FUNcube chapter summarizes the relationship between the synthetic benchmark and real-telemetry stress tests in a compact form. Table E.1 provides the detailed mapping. It separates behaviours directly supported by FUNcube morphology, behaviours only partially supported or motivated by FUNcube, behaviours that remain synthetic-only because the required labels or sensor channels are unavailable, and workflow-level support for the residual-to-event analysis process.

Table E.1: Detailed relationship between synthetic benchmark behaviours and FUNcube real-telemetry stress-test evidence.

Synthetic benchmark behaviour	FUNcube relationship	Support provided	Claim boundary
Direct morphology support from FUNcube			
Orbit-scale heating and cooling waveform	Directly visible in FUNcube temperature telemetry.	Supports the use of compact orbit-scale thermal templates as plausible nominal structure.	Does not prove flight-accurate Delfi Twin temperatures.
Spin, ripple, or illumination-imprinted thermal texture	Visible in panel ripple regimes.	Supports inclusion of residual texture and panel-level nuisance structure.	No independent attitude-state truth; not a confirmed spin anomaly.
Contact gaps and partial observability	Directly visible in gap-limited windows.	Supports explicit gap handling and treating gaps as loss of observability.	Does not validate event recall across gaps.
Observation-layer behaviour supported or motivated by FUNcube			

Continued on next page

Table E.1: Detailed relationship between synthetic benchmark behaviours and FUNcube evidence. Continued.

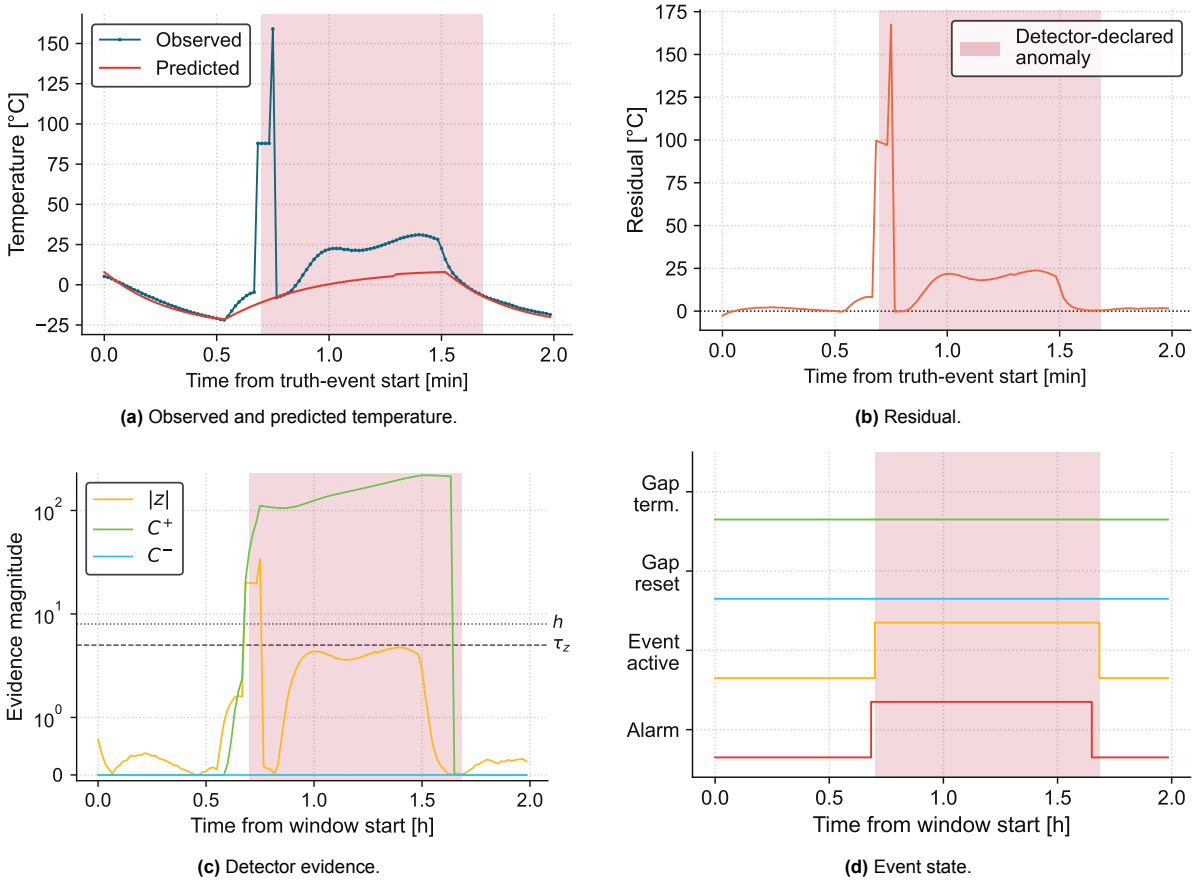
Synthetic benchmark behaviour	FUNcube relationship	Support provided	Claim boundary
Isolated spikes and abnormal telemetry segments	Visible in the telemetry-quality artifact case.	Supports spike visibility, transient-suppression logic, and separate handling of abnormal measurement behaviour.	Does not establish spike frequency, cause, or operational rate.
Pinned, held, or saturation-like telemetry behaviour	Morphologically similar behaviour is visible in selected telemetry-quality artifact windows.	Supports inclusion of alert-worthy telemetry-quality anomalies that compromise interpretation of the temperature stream.	Exact measurement-chain failure mode is not identified.
Small quantization and ordinary low-amplitude noise	Not reliably assessable from the available FUNcube warehouse data.	Retained as benign nuisances because finite resolution and residual variance can perturb detector evidence.	Absence of visible quantization or noise at the available cadence is not evidence that these effects are unrealistic.
Abrupt calibration step	Not specifically observed in the selected FUNcube windows.	Retained as a controlled telemetry-quality anomaly because sudden measurement offsets are operationally plausible and can mimic level shifts.	Synthetic stress case only; not FUNcube-confirmed.
Synthetic-only mission-scoped thermal fault behaviours			
Battery overtemperature or incipient runaway	Not directly observable with the available FUNcube channels used here.	Retained as a high-consequence Delfi Twin thermal stress case.	Requires battery or nearby internal temperature sensing; synthetic labelled fault only.
MCU or internal self-heating anomaly	Not directly observable with the available FUNcube channels used here.	Retained because subsystem activity can produce local thermal deviations in instrumented internal nodes.	Requires internal node telemetry; synthetic labelled fault only.
Heater-control failure	Not directly observable with the available FUNcube channels used here.	Retained as a mission-relevant Delfi Twin thermal fault family.	Requires heated-component telemetry and heater-state context; synthetic labelled fault only.
Thermal-interface or internal conductance degradation	Poorly observable from panel-only or externally dominated telemetry.	Retained as a predictor-validity and future multichannel-coupling concern.	Requires coupled internal and external temperature nodes; not validated by FUNcube.
Workflow-level support			
Residual-to-event workflow	Exercised on all selected FUNcube stress-test windows.	Shows that predictions, residuals, evidence traces, gaps, and detector states remain inspectable on messy real telemetry.	Qualitative workflow stress test only; not labelled detector validation.

E.2. Full FUNcube Diagnostic Stress-Test Figures

The following figures show the full diagnostic views for the FUNcube stress-test windows discussed in the main text. Each figure retains the complete processing trace: observed and predicted temperature, residual, detector evidence, and event state. These full plots are included to preserve traceability for the qualitative interpretations made in the main FUNcube chapter.

E.2.1. Telemetry-Quality Artifact Window

Figure E.1 shows the full diagnostic plot for the telemetry-quality artifact window. The case is used to support the claim that abnormal measurement behaviour remains visible in the residual and detector-evidence traces, not to identify a confirmed spacecraft fault or to estimate artifact frequency.



(a) Observed and predicted temperature.

(b) Residual.

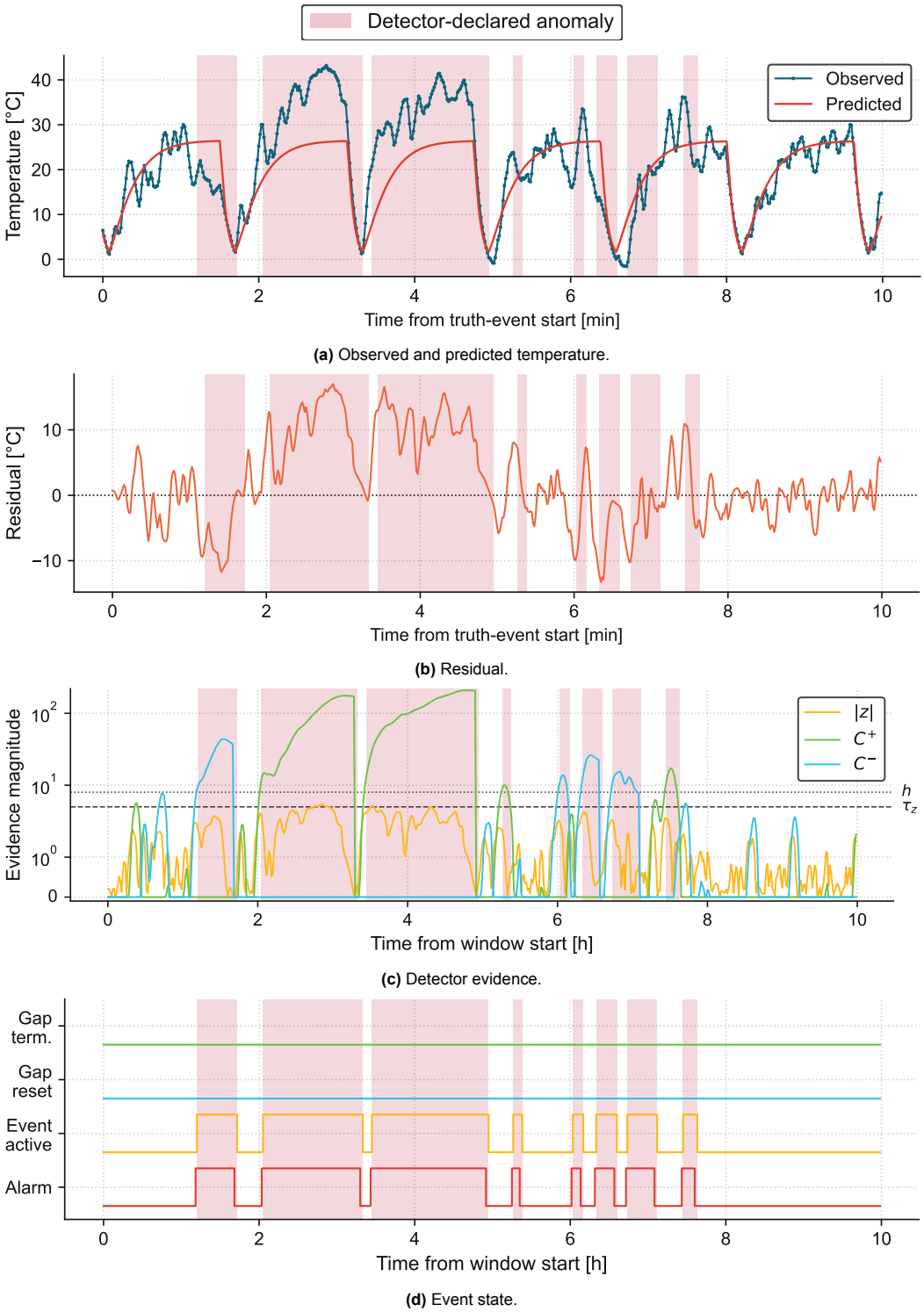
(c) Detector evidence.

(d) Event state.

Figure E.1: Full FUNcube diagnostic plot for the telemetry-quality artifact stress-test window in the Solar Panel $-Y$ temperature channel on 4 September 2014, showing temperature, residual, detector evidence, and event state.

E.2.2. Contextual Hot-Orbit Window

Figure E.2 shows the full diagnostic plot for the contextual hot-orbit window. The case illustrates how real telemetry can produce sustained contextual residual evidence when the fitted expected-temperature model does not represent the observed thermal regime. It is not interpreted as a confirmed thermal fault.



(a) Observed and predicted temperature.

(b) Residual.

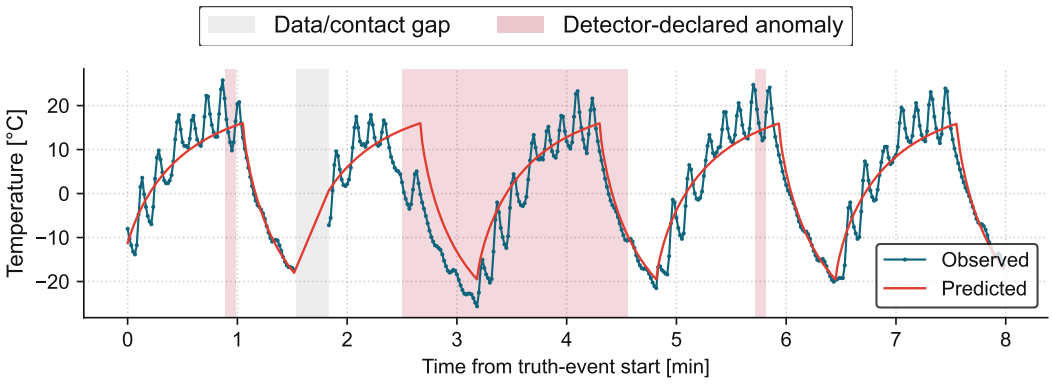
(c) Detector evidence.

(d) Event state.

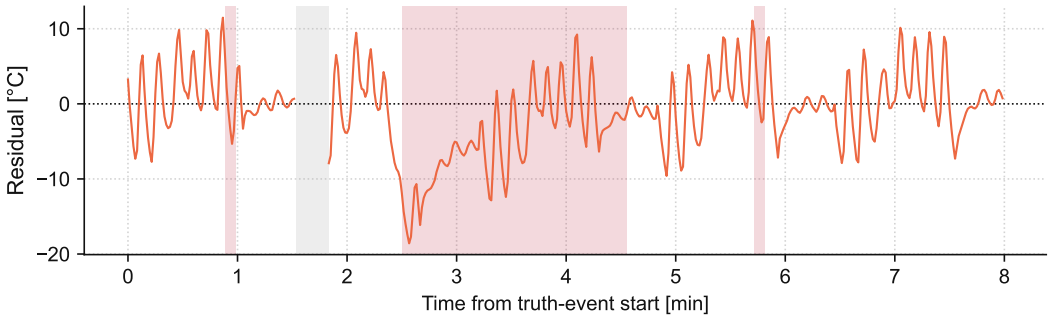
Figure E.2: Full FUNcube diagnostic plot for the contextual hot-orbit stress-test window in the silver-panel temperature channel on 12 May 2019, showing temperature, residual, detector evidence, and event state.

E.2.3. Gap-Limited Cooler-Orbit Window

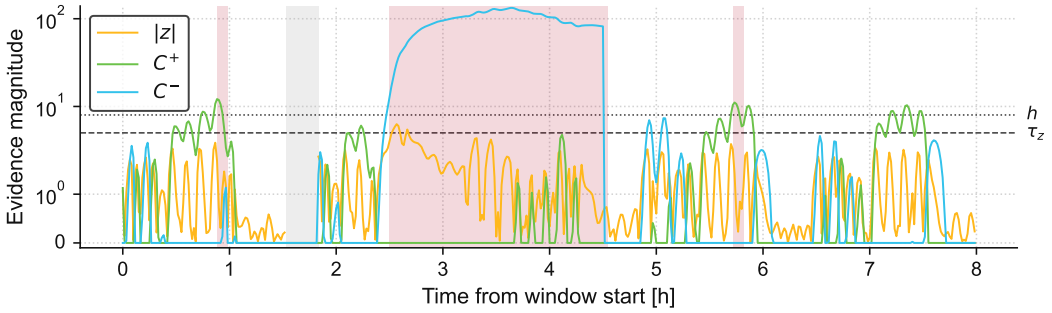
Figure E.3 shows the full diagnostic plot for the gap-limited cooler-than-expected orbit window. This case is used to illustrate how explicit observability loss is represented in the workflow. The gap handling prevents the event lifecycle from implying continuous observation across a contact gap.



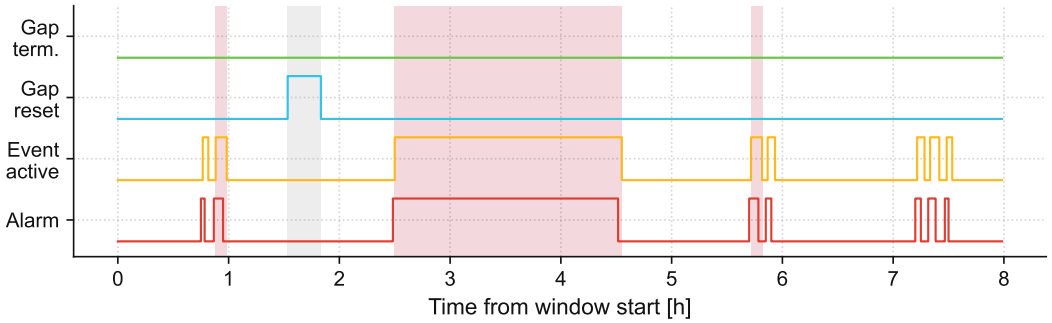
(a) Observed and predicted temperature.



(b) Residual.



(c) Detector evidence.



(d) Event state.

Figure E.3: Full FUNcube diagnostic plot for the gap-limited cooler-than-expected orbit stress-test window in the Solar Panel $-X$ temperature channel on 14 September 2015, showing temperature, residual, detector evidence, and event state.