



From Failure Reports to Grounded Diagnosis: A Review of Spacecraft Anomaly Detection and Root-Cause Analysis, and the Process Behind an Evidence-Grounded AI Framework for Lunar & Space Missions

Sangharaj Jadhav^{a,*}, Alihana Memon^a, Aditya Yadav^a, Prof. Deepti Vijay Chandran^b

^aUG Student, Department of Computer Engineering, Smt. Indira Gandhi College of Engineering (Autonomous), affiliated to University of Mumbai, Ghansoli, Navi Mumbai, Maharashtra, India

^bProject Guide, Department of Computer Engineering, Smt. Indira Gandhi College of Engineering (Autonomous), affiliated to University of Mumbai, Ghansoli, Navi Mumbai, Maharashtra, India

ABSTRACT:

Five lunar landing attempts between 2019 and 2024 — Beresheet, Hakuto-R Mission 1, Peregrine Mission One, SLIM and IM-1 Odysseus — failed or ended in off-nominal landings, and in each case the cause was established only after weeks or months of expert investigation. This review paper describes how we identified this problem, what the existing literature offers, and how a structured study of its gaps led us to the idea of LUNAR.AI, an evidence-grounded framework for detecting and explaining anomalies in spacecraft telemetry. We first analyse the publicly reported causes of the five failures and observe that each one broke a known physical relationship between sensors. We then review four bodies of work: classical fault detection, isolation and recovery (FDIR) and analytical redundancy; data-driven anomaly detection with deep learning and isolation-based methods; benchmark and evaluation critiques of time-series anomaly detection; and large language models (LLMs) for root-cause analysis with retrieval and agent architectures. From this review we derive five gaps: detectors report when telemetry is abnormal but not why; labelled failure data are scarce; raw descent telemetry is strongly non-stationary; LLM explanations are fluent but often unsupported; and evaluations rarely report uncertainty. We show how these gaps shaped five guiding principles — check physics rather than raw numbers, create the data that cannot be collected, start simple, explain like an investigation board with evidence, and measure honestly — and how the principles combine into the idea of LUNAR.AI. We then describe, at a conceptual level, how the proposed framework is intended to work, illustrate it with an example, and discuss its expected outcomes, open challenges and future scope.

Keywords: Spacecraft telemetry; Anomaly detection; Root-cause analysis; Analytical redundancy; Large language models; Retrieval-augmented generation; Lunar landing

Introduction

Space missions generate continuous streams of telemetry, yet deciding quickly that something is wrong — and, more importantly, why — remains slow, manual and dependent on a small number of experts. The recent wave of commercial and national lunar landers has made this visible. Of the five landing attempts we studied, Beresheet and Hakuto-R Mission 1 crashed, Peregrine Mission One never attempted a landing, and SLIM and IM-1 Odysseus landed but in an unintended attitude (Davis, 2019; Smith, 2023, 2024a, 2024b; Japan Aerospace Exploration Agency [JAXA], 2024). In every case the telemetry needed to recognise the problem was available on the ground, but the explanation of what had happened was produced later by engineering teams and review boards.

This paper is a review written from the perspective of a student research project. Rather than presenting the internal algorithms of a system, it documents how the idea behind it was formed: how the problem was identified from real failure reports, what existing approaches offer, where they fall short, and how each shortcoming led to a guiding principle and finally to a framework concept. We believe that making this reasoning explicit is useful to other researchers and students who want to apply artificial intelligence (AI) to safety-critical engineering problems, where the path from a broad idea to a defensible design is often the hardest part of the work.

The contributions of this review are: (i) a structured analysis of five lunar landing failures from the point of view of telemetry-based diagnosis; (ii) a review of fault detection, data-driven anomaly detection, evaluation practice and LLM-based root-cause analysis; (iii) a gap analysis that maps each limitation to a guiding principle; and (iv) a conceptual description of the resulting framework, LUNAR.AI, and how it is intended to work. The overall research process is shown in Fig. 1.

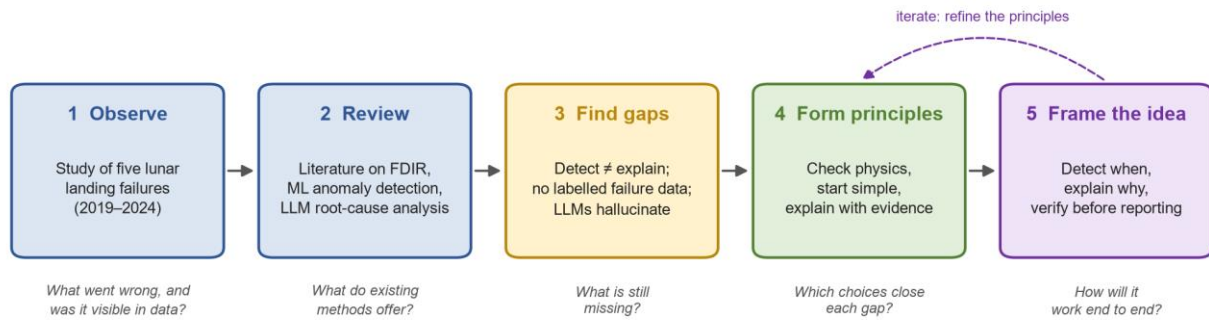


Fig. 1 - Research process followed in this work, from studying real failures to a framework concept.

How the Problem Was Identified

1.1. Starting point: a broad question

The project began with a broad question: can AI help engineers understand spacecraft failures faster? This question is too wide to answer directly, so we narrowed it in three steps. First, we chose a domain with recent, well-documented, publicly reported failures — lunar landing — so that our ideas could be checked against real events. Second, we read the official statements and reputable reporting on each failure and recorded, for every mission, the publicly stated cause and the sensors that would have been involved. Third, we asked a falsifiable question: was the fault, in principle, visible in the telemetry before the mission was lost?

1.2. Analysis of five lunar landing failures

Table 1 summarises what we found. Beresheet (SpaceIL, 2019) lost its main engine after a command intended to recover a faulty inertial measurement unit triggered a chain of computer resets (Davis, 2019). Hakuto-R Mission 1 (ispace, 2023) ran out of propellant above the surface because its software rejected valid altimeter data after flying over a crater rim and therefore believed it was lower than it was (Smith, 2023). Peregrine Mission One (Astrobotic, 2024) suffered a propellant loss after a pressure control valve failed (Smith, 2024b). SLIM (JAXA, 2024) lost one of its two main engines at about 50 m altitude and came to rest in an unintended attitude (JAXA, 2024). IM-1 Odysseus (Intuitive Machines, 2024) descended without its laser rangefinders and came to rest on its side (Smith, 2024a).

Table 1 - Publicly reported causes of five lunar landing failures and the physical relationship each one disturbed.

Mission (year)	Outcome	Publicly reported cause	Physical relationship disturbed
Beresheet (2019)	Crash	Fault-recovery command for an IMU led to computer resets and main-engine shutdown	Thrust and chamber pressure no longer follow the throttle command
Hakuto-R M1 (2023)	Crash	Software rejected valid altimeter data; propellant exhausted while still above the surface	Measured altitude change disagrees with estimated vertical velocity
Peregrine (2024)	Landing abandoned	Pressure control valve failure caused loss of propellant	Propellant mass falls faster than the engine consumes it
SLIM (2024)	Landed, wrong attitude	One of two main engines lost at about 50 m altitude	Acceleration falls short of that expected from commanded thrust
IM-1 Odysseus (2024)	Landed on its side	Descent without laser rangefinders; excess velocity at touchdown	Altitude and velocity estimates become inconsistent

1.3. Key observation

Reading the cases side by side produced the central observation of this work: every failure broke a known physical relationship between two or more sensors — thrust against throttle command, propellant mass against engine flow, altitude rate against vertical velocity. The difficulty was not that the data were missing, but that the relationships had to be checked, interpreted and connected to a cause by people. This reframed the problem from "detect unusual numbers" to "detect broken physics and explain which physics broke, with evidence". The rest of the project followed from this reframing.

Review of Existing Approaches

We organised the literature around the two halves of the reframed problem — detection and explanation — and added a third strand on how such methods are evaluated. Table 2 compares representative works.

1.4. Model-based fault detection and analytical redundancy

The oldest and most trusted family of methods in aerospace is model-based fault detection. Chow and Willsky (1984) formalised analytical redundancy: instead of duplicating hardware, the relationships implied by a system model are used to compute residuals that stay near zero in normal operation and deviate when a fault occurs. Isermann (2005) and Venkatasubramanian et al. (2003) review how residual generation, evaluation and diagnosis are applied in industry. These methods are interpretable, because each residual corresponds to a physical law, and they need no failure data. Their weaknesses are that the model must be built by hand, thresholds are tuned manually, and the output is an alarm rather than an explanation that a non-specialist can act on.

1.5. Data-driven anomaly detection

Machine learning approaches learn normal behaviour from data. Chandola et al. (2009) survey the classical landscape. For spacecraft, Hundman et al. (2018) predicted NASA SMAP and MSL telemetry with LSTM networks and flagged anomalies with a nonparametric dynamic threshold, and Malhotra et al. (2016) used an LSTM encoder-decoder to reconstruct multi-sensor signals. Isolation Forest (Liu et al., 2008) isolates anomalies with random partitions and is fast and unsupervised, while the Anomaly Transformer (Xu et al., 2022) uses attention-based association discrepancy. Mansell and Spencer (2021) applied deep learning to fault diagnosis for spacecraft attitude determination and control and could classify fault types. Physics-informed neural networks (Raissi et al., 2019) show that physical laws can be built into learning. Across this work, however, the output is a score or a class label: the detectors answer when telemetry became abnormal, but not why, and none cites evidence for its conclusion.

1.6. Evaluation practice

Two critiques strongly shaped our thinking. Wu and Keogh (2023) showed that many popular time-series anomaly benchmarks are flawed — some anomalies are trivially detectable and labels are sometimes wrong — so reported progress can be an illusion. Kim et al. (2022) showed that the common point-adjustment protocol inflates F1 scores so much that random scores can appear competitive. The lesson for us was that a complex model should never be adopted without comparing it with a simple baseline under an honest protocol, and that results on small datasets must be reported with uncertainty (Efron & Tibshirani, 1993).

1.7. Large language models for root-cause analysis

Explaining an anomaly is a reasoning task, and LLMs have recently been applied to it. Ahmed et al. (2023) evaluated LLMs for recommending root causes and mitigation steps for cloud incidents at Microsoft and found them useful but imperfect. Retrieval-augmented generation (RAG) conditions a model on retrieved documents to improve factual accuracy (Lewis et al., 2020), with lexical retrieval such as BM25 (Robertson & Zaragoza, 2009) remaining a strong, transparent choice. Chain-of-thought prompting (Wei et al., 2022) and the ReAct pattern of interleaved reasoning and acting (Yao et al., 2023) structure the reasoning process, and debate between multiple agents can improve factuality (Du et al., 2024). Ma et al. (2024) combined LLMs with physical simulation for scientific discovery. Against these advances stands the problem of hallucination: LLMs can produce fluent, confident statements that are not supported by their input (Ji et al., 2023). In a safety-critical setting, an unsupported explanation is worse than none.

Table 2 - Comparison of representative approaches against the needs identified in Section 2 (Y = yes, P = partly, N = no).

Work	Detects	Explains cause	Cites evidence	Needs no failure labels
Chow & Willsky (1984)	Y	P	N	Y
Hundman et al. (2018)	Y	N	N	Y
Liu et al. (2008)	Y	N	N	Y
Xu et al. (2022)	Y	N	N	Y
Mansell & Spencer (2021)	Y	P	N	N
Ahmed et al. (2023)	N	Y	N	N
Lewis et al. (2020)	N	P	Y	Y
Proposed idea (target)	Y	Y	Y	Y

Gap Analysis

Combining the failure study with the literature review, we identified five gaps. Each gap is stated together with the evidence that revealed it and the design decision it led to (Table 3).

- G1 — Detection without explanation. Detectors report when telemetry is abnormal but not why (Hundman et al., 2018; Liu et al., 2008); the failure reports show that the why is what investigators need.
- G2 — Scarcity of labelled failure data. Real lander failures are rare, and their full telemetry is rarely published, so supervised learning on real failures is not feasible.
- G3 — Non-stationary telemetry. During a descent, altitude falls from kilometres to zero and velocity, mass and throttle all change, so a detector trained on raw channels mostly learns where in the descent the vehicle is.

- G4 — Ungrounded explanations. LLMs can explain, but without evidence and checks their explanations may be hallucinated (Ji et al., 2023).
- G5 — Weak evaluation. Benchmarks can mislead and small studies are often reported without uncertainty (Wu & Keogh, 2023; Kim et al., 2022).

Table 3 - Mapping from identified gaps to design decisions.

Gap	Evidence	Design decision
G1 Detect, not explain	Failure reports; detector outputs are scores	Separate investigation stage that produces a cited, verified explanation
G2 No failure data	Few public failure datasets	Physics simulator with fault injection, plus reconstructions of real failures
G3 Non-stationary data	Descent profile; analytical redundancy	Check consistency with physics instead of raw values
G4 Hallucination	Ji et al. (2023); RAG literature	Every explanation must point to evidence and pass an independent check
G5 Weak evaluation	Wu & Keogh (2023); Kim et al. (2022)	Start simple; evaluate honestly and report uncertainty

How the Idea Was Formed

The gaps did not point to a single algorithm; they pointed to a set of principles that any credible solution would have to respect. This section explains how each principle emerged and which alternatives we set aside along the way.

1.8. Principle 1: Check physics, not just numbers (G3, G1)

Our first instinct, like much of the literature, was to let a learning model look at the raw telemetry and flag whatever seemed unusual. Thinking about a descent quickly showed why this is fragile: almost every quantity changes continuously during a landing, so "unusual" is hard to define. The failure study (Section 2.3) suggested a different question — not "is this number strange?" but "is the spacecraft still obeying the physics it was designed to obey?". The classical idea of analytical redundancy (Chow & Willsky, 1984) gave this question a sound footing: what the sensors measure can be compared with what the vehicle's physics says they should measure, and the difference stays small while the system is healthy. An important side benefit is explainability — when such a comparison fails, it already tells us which physical relationship broke, which is the first step towards answering why.

1.9. Principle 2: Create the data we cannot collect (G2)

Any method needs examples of failures to be developed and tested, and real lander failures are, fortunately, rare. Rather than wait for data that may never be published, we decided that the idea must include a physics-based simulation of the flight in which faults of known type and timing can be introduced deliberately, so that the correct answer is always known. To keep the work anchored in reality, the same simulation is used to reconstruct, in simplified form, the publicly reported failures from Table 1. We preferred physics-based simulation over purely data-driven generation of synthetic data, because only the former guarantees that the generated behaviour obeys physical laws and carries a known cause (Raissi et al., 2019, show the value of building physics into learning).

1.10. Principle 3: Start simple, add complexity only when justified (G5)

The evaluation critiques of Wu and Keogh (2023) and Kim et al. (2022) changed how we approached model selection. Instead of beginning with the most sophisticated model available, the idea is to begin with the simplest method that could work, compare it fairly with more complex alternatives, and adopt complexity only when it brings a clear and measurable benefit. In a safety-critical setting this also has a practical advantage: simpler methods are easier for engineers to understand, inspect and trust.

1.11. Principle 4: Explain like an investigation board, with evidence (G1, G4)

Detection tells engineers when something went wrong; they also need to know why. We looked at how real failures are explained and noticed that review boards work as teams with distinct roles: someone establishes the facts from the data, others propose possible causes, others search records and manuals for supporting or contradicting evidence, and finally the board decides which explanation holds. Large language models are well suited to the reasoning parts of this work (Wei et al., 2022; Yao et al., 2023), and several cooperating agents can check one another (Du et al., 2024). Because of the hallucination problem (Ji et al., 2023), however, the idea adds two safeguards: explanations must be grounded in retrieved engineering knowledge (Lewis et al., 2020), and no explanation is reported as supported unless the evidence it refers to actually exists and has passed an independent check. The language model therefore assists the investigation, but it is not trusted to decide on its own.

1.12. Principle 5: Measure honestly (G5)

Finally, we decided how the idea should be judged before building it. Because the correct cause is known for every simulated fault, the framework can be assessed on whether it detects the fault, how quickly, how often it raises false alarms, whether it names the right cause, and whether its explanations are actually supported by evidence. Because the number of test cases will be small, results should be reported with confidence intervals (Efron & Tibshirani, 1993) rather than as single numbers.

Proposed Framework: How It Will Work

1.13. Overview

The five principles combine into the concept of LUNAR.AI shown in Fig. 2. Telemetry from the spacecraft first passes through a physics consistency stage that compares measured behaviour with the behaviour expected from the vehicle's design. An anomaly detection stage watches these comparisons and decides when the flight departs from normal. When it does, an evidence-grounded investigation stage takes over to work out why: it gathers the facts, proposes possible causes and searches a knowledge base of engineering documents for evidence. A verification stage then checks every claim; claims without valid evidence are rejected and returned to the investigation. The result is an incident report that states the most likely cause, the evidence for it and how confident the system is. A physics simulation supports the whole framework by supplying fault scenarios and reconstructions of real failures for development and testing.

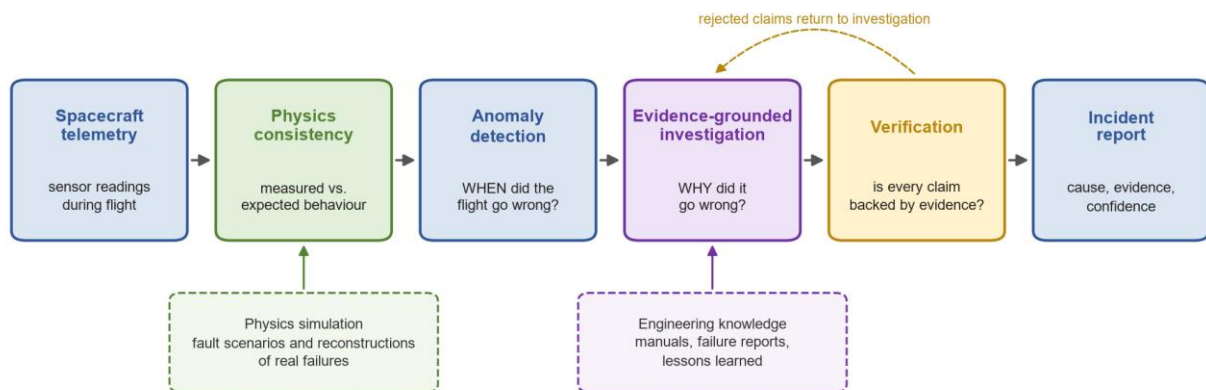


Fig. 2 - Conceptual view of the proposed LUNAR.AI framework: detect when, explain why, and verify before reporting.

1.14. An illustrative example

Consider a lander whose propellant tank develops a leak during descent, similar in nature to the problem reported for Peregrine (Smith, 2024b). To the eye, the telemetry may look almost normal, because propellant is expected to decrease during a burn anyway. The physics consistency stage, however, notices that propellant is disappearing faster than the engine can be consuming it, and the anomaly detection stage marks the moment this begins. The investigation stage then records the facts, proposes candidate causes — for example a leak, a faulty fuel gauge or an engine running richer than commanded — and looks for evidence for and against each one. The verification stage accepts the leak hypothesis only if the evidence supports it and rejects the others with reasons. Instead of an alarm that a specialist must interpret, the engineer receives an explanation that can be checked step by step.

1.15. Intended users and use

The framework is intended as a decision-support tool for mission engineers and failure investigators, and as a learning tool for students who want to understand how spacecraft fail. It does not replace human judgement; its purpose is to shorten the time between noticing a problem and understanding it, and to make every step of its reasoning open to inspection.

Expected Outcomes and Challenges

If the idea works as intended, we expect three outcomes: faults detected early and with few false alarms, because detection is based on physical consistency rather than raw values; explanations that engineers can trace back to data and documents, because unsupported claims are filtered out; and a reusable way of studying past failures through simulation. We also expect that retrieval of engineering knowledge will matter less for guessing the faulty subsystem than for making the explanation verifiable — a hypothesis worth testing explicitly.

Several challenges remain open. A simulation is only as realistic as its model, and reconstructions built from public reports are simplified. Real flight telemetry is rarely published, which limits validation. Language models change quickly and may behave differently across versions, so the

safeguards must not depend on any one model. Finally, the quality of an explanation is ultimately judged by experts, so their involvement will be essential in later stages of the work.

Conclusion and Future Scope

This review traced how the idea behind LUNAR.AI was formed. A study of five lunar landing failures showed that each one broke a known physical relationship between sensors; a review of the literature showed that existing detectors do not explain, that language-model explanations need grounding, and that evaluations are often weaker than they appear. These findings led to five principles — check physics, create the data that cannot be collected, start simple, explain with evidence, and measure honestly — and to a framework that detects when a flight goes wrong, explains why, and verifies every claim before reporting it. Future work will develop and evaluate the framework against these principles, extend it from lunar landing to other mission phases such as launch, test it on public spacecraft telemetry (Hundman et al., 2018), and involve domain experts in assessing the explanations it produces.

Acknowledgements

We thank our project guide, Prof. Deepti Vijay Chandran, for her guidance throughout this work, and Dr. K. T. Patil, Head of the Department of Computer Engineering, and Dr. Sunil Chavan, Principal, Smt. Indira Gandhi College of Engineering, for their support.

REFERENCES

- Ahmed, T., Ghosh, S., Bansal, C., Zimmermann, T., Zhang, X., & Rajmohan, S. (2023). Recommending root-cause and mitigation steps for cloud incidents using large language models. In *Proceedings of the 45th IEEE/ACM International Conference on Software Engineering (ICSE)* (pp. 1737–1749). IEEE.
- Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, 41(3), Article 15, 1–58.
- Chow, E. Y., & Willsky, A. S. (1984). Analytical redundancy and the design of robust failure detection systems. *IEEE Transactions on Automatic Control*, 29(7), 603–614.
- Davis, J. (2019, April 17). Software command to fix faulty sensor doomed Beresheet. The Planetary Society. <https://www.planetary.org/articles/software-command-doomed-beresheet>
- Du, Y., Li, S., Torralba, A., Tenenbaum, J. B., & Mordatch, I. (2024). Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning* (Vol. 235). PMLR.
- Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. Chapman & Hall.
- Hundman, K., Constantinou, V., Laporte, C., Colwell, I., & Soderstrom, T. (2018). Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 387–395). ACM.
- Isermann, R. (2005). Model-based fault-detection and diagnosis – status and applications. *Annual Reviews in Control*, 29(1), 71–85.
- Japan Aerospace Exploration Agency. (2024, January 25). Outcome for the Smart Lander for Investigating Moon (SLIM)'s moon landing [Press release]. https://global.jaxa.jp/press/2024/01/20240125-1_e.html
- Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., & Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12), Article 248, 1–38.
- Kim, S., Choi, K., Choi, H.-S., Lee, B., & Yoon, S. (2022). Towards a rigorous evaluation of time-series anomaly detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(7), 7194–7201.
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33, 9459–9474.
- Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM)* (pp. 413–422). IEEE.
- Ma, P., Wang, T.-H., Guo, M., Sun, Z., Tenenbaum, J. B., Rus, D., Gan, C., & Matusik, W. (2024). LLM and simulation as bilevel optimizers: A new paradigm to advance physical scientific discovery. In *Proceedings of the 41st International Conference on Machine Learning* (Vol. 235, pp. 33940–33962). PMLR.
- Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., & Shroff, G. (2016). LSTM-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*.
- Mansell, J. R., & Spencer, D. A. (2021). Deep learning fault diagnosis for spacecraft attitude determination and control. *Journal of Aerospace Information Systems*, 18(3), 102–115.
- Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378, 686–707.
- Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4), 333–389.
- Smith, M. (2023, May 26). Japanese lunar lander Hakuto-R M1 failed due to software error. *SpacePolicyOnline.com*. <https://spacepolicyonline.com/?p=62380>
- Smith, M. (2024a, February 23). IM's Odysseus lunar lander not upright after all. *SpacePolicyOnline.com*. <https://spacepolicyonline.com/news/ims-odysseus-lunar-lander-not-upright-after-all/>

-
- Smith, M. (2024b, August 27). Independent review confirms bad valve doomed Astrobotic's Peregrine-1 mission. SpacePolicyOnline.com. <https://spacepolicyonline.com/?p=73769>
 - Venkatasubramanian, V., Rengaswamy, R., Yin, K., & Kavuri, S. N. (2003). A review of process fault detection and diagnosis: Part I: Quantitative model-based methods. *Computers & Chemical Engineering*, 27(3), 293–311.
 - Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q. V., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35, 24824–24837.
 - Wu, R., & Keogh, E. J. (2023). Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *IEEE Transactions on Knowledge and Data Engineering*, 35(3), 2421–2429.
 - Xu, J., Wu, H., Wang, J., & Long, M. (2022). Anomaly Transformer: Time series anomaly detection with association discrepancy. In *International Conference on Learning Representations (ICLR)*.
 - Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.