Chronos-2 for Spacecraft Telemetry Forecasting

Žiga Klun
Bias Variance Labs d.o.o.
Ljubljana, Slovenia
zk56034@student.uni-lj.si

Dragi Kocev
Bias Variance Labs d.o.o.
Jožef Stefan Institute
Ljubljana, Slovenia

Jure Brence
Bias Variance Labs d.o.o.
Jožef Stefan Institute
Ljubljana, Slovenia

Abstract

Spacecraft telemetry is continuously monitored for anomalies, and the dominant automated approach forecasts nominal behaviour and applies a threshold to the resulting residuals, so the detection quality is affected by the forecast quality. We compare the per-channel LSTM inspired by Telemanom against Chronos-2, a pretrained model supporting multivariate and covariate-informed forecasting, on 76 channels of the ESA Anomaly Dataset. The best results come from Chronos-2 fine-tuned with LoRA, followed closely by the same model applied in zero-shot mode. In both cases, Chronos-2 reduces the median normalised mean squared error by over 50% compared to the strongest LSTM baseline. The small improvement from fine-tuning indicates that most of Chronos-2's advantage is available without mission-specific adaptation. Supplying telecommands as covariates does not meaningfully affect the performance of Chronos-2, which we attribute to the exclusion of rare events from our evaluation windows.

Keywords

spacecraft telemetry, time series forecasting, foundation models, zero-shot forecasting, anomaly detection, space operations

1 Introduction

Spacecraft continuously downlink telemetry in the form of time series that record temperatures, voltages, currents, attitude, and the state of on-board instruments. A single satellite can produce recordings from thousands of such channels, and monitoring them is a daily responsibility of operations engineers in mission control centres [4, 7]. The consequences of missing a fault are severe, since a spacecraft cannot be repaired once launched and a problem that goes unnoticed can escalate into the loss of an instrument or of the entire mission. At the same time, the volume of downlinked data keeps growing, which makes manual inspection increasingly impractical.

Operational anomaly detection still relies mostly on out-of-limits checks combined with manual analysis [4, 7], which is blind to anomalies that stay inside the nominal limits but break the expected temporal or cross-channel pattern. The dominant data-driven alternative is *forecasting-based* detection, in which a model trained on nominal telemetry predicts how each channel should continue and flags an anomaly when the prediction error is high enough. Detection quality is therefore affected by forecast quality, since a forecaster that cannot represent nominal dynamics produces large residuals during nominal operation, and those become false alarms.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.104

The canonical instance of this design is Telemanom [4], which trains one LSTM per channel and applies an unsupervised dynamic threshold to the resulting prediction errors. It remains the best performing algorithm on Mission 1 of the ESA Anomaly Detection Benchmark (ESA-ADB) [7], and is therefore the baseline we compare against. Two of its properties motivate this work. It forecasts one channel at a time, so cross-channel interactions are unused, and it produces a point prediction with no measure of its own uncertainty, so a separate stage is needed to decide which errors are large enough to matter.

Transformer-based foundation models, pretrained once and applied across forecasting and anomaly detection alike, have meanwhile become an important paradigm in time series analysis [8]. Chronos-2 [1], a recent model of this kind, addresses both limitations. It performs univariate, multivariate and covariate-informed forecasting zero-shot, predicts a long horizon in one pass, and emits quantiles. We therefore isolate the forecasting component, which can be evaluated without committing to any thresholding scheme, and compare an LSTM baseline against Chronos-2 on Mission 1 of ESA-ADB [7], under several input configurations, zero-shot and LoRA fine-tuned, and with telecommands as covariates.

2 Related Work

2.1 Anomaly detection in spacecraft telemetry

Expert systems, nearest neighbour searches and clustering have all been deployed operationally, yet out-of-limits checking remained the most widely deployed approach because of its low cost and interpretability [4]. Deep learning shifted the field towards forecasting-based detection, in which a network trained on nominal data acts as a predictor and its prediction errors are scored for anomalousness [12]. Its merits against classification-based alternatives have since been measured directly [10]. Telemanom [4] pairs per-channel LSTM forecasting with unsupervised dynamic thresholding of the prediction errors. Benecki et al. [2] showed that this thresholding stage is itself highly tunable, and Lakey and Schlippe compared recurrent, convolutional and transformer architectures for the task [9]. Beyond forecasting-based methods, reconstruction-based anomaly detection uses autoencoders to compress nominal telemetry and score the reconstruction error [6, 16]. ESA-ADB benchmarks one such model, DC-VAE-ESA, but forecasting-based Telemanom remains the strongest performer on Mission 1 [7]. Progress was long limited by a lack of realistic labelled data. The SMAP and MSL sets released with Telemanom [4] were the de facto standard for years, until Wu and Keogh [15] showed that these and other popular benchmarks suffer from flaws severe enough to make much of the reported progress illusory. OPS-SAT [13] added annotated telemetry from an ESA CubeSat, and ESA-ADB [7] is the most recent and by far the largest, with orders of magnitude more data points than any earlier public satellite telemetry dataset.

2.2 Time series forecasting

Classical methods such as ARIMA and exponential smoothing are *local*, fitting one set of parameters per series [5], while deep learning introduced *global* models sharing parameters across series, including DeepAR [14], which produces probabilistic rather than point forecasts, and the Temporal Fusion Transformer [11], which handles known-future and past-only covariates. Pretrained models form a third paradigm, trained once and applied to unseen data without task-specific training [1]. Most are univariate and ignore covariates, which limits their use on telemetry, where channels are coupled and contextual data, such as telecommands, can carry critical information.

3 Experimental Setup

3.1 The ESA Anomaly Benchmark

ESA-ADB is a benchmark for anomaly detection in satellite telemetry data, assembled by ESA operations engineers together with machine learning researchers from real, manually annotated telemetry of three ESA missions [7]. We use Mission 1 of ESA-ADB, the longest-running of its three missions. It holds 76 telemetry channels recorded over an anonymised span of about 14 years, some 775 million points, annotated by operations engineers with 200 events of which 118 are anomalies and 78 are rare nominal events such as commanded manoeuvres. The data is kept close to what operators actually see, including irregular timestamps, varying sampling rates and communication gaps.

3.2 Forecasting task construction

ESA-ADB defines an anomaly detection task. As our object of study is the forecaster, we derive a forecasting task from the same data so that every model sees identical inputs and is scored on identical targets.

All 76 channels are resampled to a common 1-minute grid to address the irregular timestamps of the raw telemetry. Observations within each one-minute bin are averaged and empty bins linearly interpolated without a maximum-gap cutoff. From this grid we cut fixed windows of $C = 2,048$ context steps followed by $H = 256$ horizon steps, C matching Chronos-2's pretraining context length, so a model observes roughly 34 hours of history and predicts the next 4.3 hours, with candidate start times advanced in steps of 3 hours. A window is accepted only if neither its context nor its horizon overlaps an annotated event on any of the 76 channels, irrespective of the model inputs, so that neither anomalies nor rare nominal events leak into inputs or targets. This yields 20,000 windows, split chronologically into 16,000 training, 2,000 validation and 2,000 test, which keeps future information out of training. At each split boundary, 13 windows were purged, separating retained origins by 2,520 minutes and preventing cross-split overlap.

Interpolated horizon values are scored, while each context is independently Gaussian-filtered. Each channel is then standardised using a fixed mean and standard deviation computed over the training period. Figure 1 shows one such window, with the context the model receives and the horizon it must predict. Per-window standardisation is avoided because, on near-constant channels, local variation can be far smaller than overall variation and thereby inflate physically negligible errors.

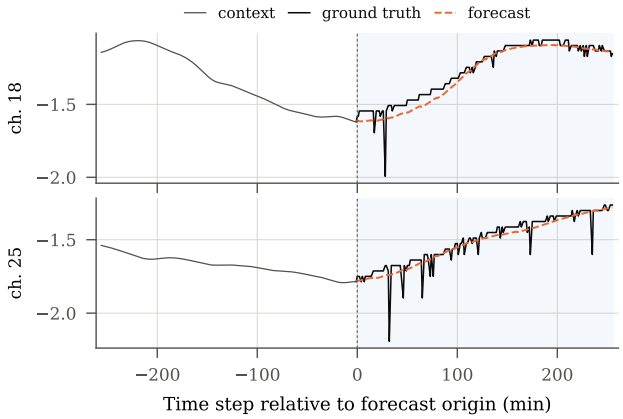

Figure 1: One test window forecast jointly by Chronos-2 (LoRA, multivariate), two of the 76 channels shown. The model sees only the context left of the dashed line. The context is Gaussian-filtered, the horizon is not. Only the final 256 of the 2048 context steps are shown.

3.3 Telecommands

The dataset also records the 698 telecommands sent to the spacecraft. These are natural explanatory inputs, since many sharp telemetry transitions are the intended consequence of a command rather than spontaneous behaviour. Following the ESA-ADB convention for full-channel experiments we keep only the commands of the highest priority. Each is a sparse event log, encoded on the telemetry grid as a binary indicator that is one in any interval containing an execution. These columns are passed to the models unchanged, without standardisation or smoothing.

4 Models

In our experiments, we compare two forecasting models.

- The **LSTM** is a Telemanom-inspired forecasting baseline [4]. It retains the two-layer recurrent architecture but replaces Telemanom's output head with a linear head that emits the complete horizon directly. We evaluate the original per-channel configuration together with fixed-capacity multivariable and multivariate variants; these are controlled configuration comparisons, not independently optimised multivariate LSTMs.
- **Chronos-2** [1] is a pretrained transformer that performs univariate, multivariate and covariate-informed forecasting in context, without task-specific training, and can forecast all 76 channels jointly. It distinguishes past-only covariates, whose values stop at the forecast origin, from known-future covariates, whose values extend across the horizon. Telecommands are known-future, since the command schedule is planned in advance. It is also probabilistic, predicting 21 quantiles from 0.01 to 0.99, so a predictive interval is supplied directly rather than reconstructed from smoothed residuals [2]. As the LSTM is a point forecaster, we use only the median quantile. We evaluate Chronos-2 zero-shot and fine-tuned with LoRA [3], which freezes the pretrained weights and injects trainable low-rank matrices.

4.1 Input configurations

The models are compared under three input configurations. In the *single-channel* configuration (SC) a model sees and predicts one channel only, reproducing Telemanom's original design. The *multivariate* configuration (MVT) gives one model all 76 channels as input and requires it to forecast all 76 jointly, so cross-channel structure is available for every target. The *multivariable* configuration (MVB) also takes all 76 channels as input but forecasts a single target channel, separating the benefit of extra context from the burden of predicting many outputs at once. Telecommands are added as extra input-only covariates on top of the multivariate configuration, either past-only or known-future, and are never forecast themselves.

5 Evaluation

The LSTM uses two layers with 64 hidden units, close to Telemanom's two-layer, 80-unit architecture, and is trained with Adam at a learning rate of 10^{-3}, a batch size of 64 and early stopping on the validation split. We hold this capacity fixed across configurations to isolate the effect of input and output dimensionality; because it was not tuned for 76 outputs, the MVT result may partly reflect a capacity bottleneck. Chronos-2 is used in its base 120M-parameter form, either zero-shot or adapted with LoRA of rank 16 and $\alpha = 32$ at a learning rate of 10^{-5} for 2000 steps. All runs were executed on a single NVIDIA B200 GPU.

We report mean squared error, mean absolute error and the coefficient of determination R^2, all in the normalised space. Each metric is computed per channel over the test windows and then aggregated across the 76 channels. We report both the mean and the median across channels, but treat the median as the headline statistic, because a handful of near-constant channels carry very large normalised errors and dominate any average. The gap between the two is itself informative, and we examine it in Section 7. We also track how error accumulates over the horizon. For each channel the squared error is averaged over test windows at each of the 256 horizon steps and then accumulated as a running mean over steps $1..k$, so the quantity at k is the mean error of a k-step forecast. These per-channel curves are aggregated by their median, and at $k = 256$ the value reproduces the horizon-averaged MSE.

6 Results

Table 1 reports every configuration over all 76 channels and Figure 2 the corresponding per-channel spread.

Chronos-2 has lower MSE and MAE than the evaluated LSTM baselines. Its best setting reaches a median MSE of 0.0021 against 0.0059 for the strongest LSTM baseline, a factor of 2.8, and the gap in median R^2 is 0.996 against 0.984. The more striking comparison is the zero-shot row: without task-specific training on this mission, Chronos-2 attains a median MSE of 0.0025, 2.3 times lower than an LSTM trained on it directly. Fine-tuning helps, but only modestly, improving median MSE from 0.0025 to 0.0021 and mean R^2 from 0.745 to 0.759. Almost all of the benefit is therefore already present in the pretrained model.

For the evaluated 64-unit LSTM, moving from separately trained single-channel models to one joint 76-output model raises median MSE from 0.0059 to 0.1968. This shows that the fixed-capacity baseline does not scale successfully to the joint-output setting, but should not be read as a general limitation of tuned multivariate LSTMs. Chronos-2 shows no analogous penalty in the configurations tested and obtains its lowest error while forecasting all 76 channels jointly.

Table 1: All configurations over the 76 channels, horizon-averaged, in normalised units.

	MSE		MAE		R^2	
Configuration	Mean	Median	Mean	Median	Mean	Median
Chronos-2 zero-shot, MVT	0.103	0.0025	0.121	0.0309	0.745	0.995
Chronos-2 LoRA, MVT	0.100	0.0021	0.114	0.0259	0.759	0.996
+ telecommands, past-only	**0.100**	0.0021	**0.113**	**0.0257**	**0.760**	**0.996**
+ telecommands, known-future	0.100	**0.0021**	0.113	0.0258	0.759	0.996
LSTM SC	0.482	0.0059	0.289	0.0515	0.759	0.984
LSTM MVT	1.263	0.1968	0.648	0.3819	0.210	0.592
+ telecommands	1.448	0.1666	0.626	0.3521	−0.592	0.670

Table 2: Horizon-averaged MSE ($\times 10^{-3}$, normalised) on the four restricted-target channels.

		LSTM			LoRA Chronos-2		
Channel	STD	MVB	MVT	SC	MVB	MVT	SC
69	0.0005	0.280	1.42	0.251	**0.246**	0.247	0.264
26	0.0396	49.9	532.8	15.9	**2.79**	2.85	3.83
14	0.0595	132.7	196.4	1.11	**1.01**	1.01	1.01
61	0.3540	6.93	370.5	0.0132	0.0122	0.0124	**0.0094**
Mean		47.4	275.3	4.31	**1.01**	1.03	1.28

The cumulative horizon-averaged error separates the models further (Figure 3). From $k = 1$ to $k = 256$, the running-mean MSE over steps $1, \ldots, k$ increases by a factor of 1.2 for Chronos-2 LoRA, 1.5 for the zero-shot model and 2.2 for LSTM SC.

Telecommands do not meaningfully affect the performance of Chronos-2. Supplying the highest-priority commands as covariates produces only very small differences in Table 1, and this holds whether they are given as past-only or as known-future covariates, even though the latter reveals the exact future command schedule. For the LSTM the effect is mixed: the median R^2 improves from 0.592 to 0.670, while the mean R^2 falls from 0.210 to −0.592 because the channels it already handled worst degrade further.

Table 2 compares all three input configurations on a small set of four channels rather than all 76, since the multivariable and single-channel settings each require a separately trained model per target. The four channels are selected to represent different standard deviations (STD), with `channel_69` having one of the smallest and `channel_61` one of the largest. Restricting the LSTM from 76 outputs to one improves its mean MSE from 275.3 to 47.4×10^{-3}, and narrowing the input improves it further to 4.31×10^{-3}. The same output restriction leaves Chronos-2 unchanged at 1.01×10^{-3}, and its single-channel result (1.28×10^{-3}) is worse. Even at its best the LSTM remains roughly 4.2 times behind.

7 Discussion

Fine-tuning helps, but the pretrained model already carries most of the benefit. In this temporal split, LoRA reduces median MSE by 17.6%, a modest observed gain whose variability was not assessed, whereas the zero-shot model, which has never seen this mission, already outperforms an LSTM trained directly on it. Operationally this matters more than the ranking itself. Telemanom requires one trained model per channel, and those models must be maintained as the spacecraft ages and its behaviour drifts, whereas a pretrained model can be applied to a new channel, or

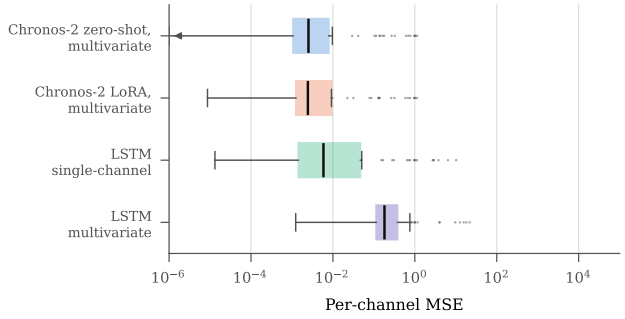

Figure 2: Per-channel horizon-averaged MSE, one box per configuration. The axis is truncated at 10^{-6}; the arrow indicates eight zero-shot values below the display floor, the smallest approximately 2×10^{-15}.

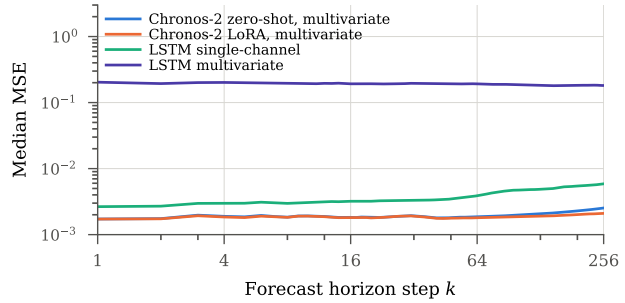

Figure 3: Median across the 76 channels of cumulative MSE over forecast steps $1, \ldots, k$.

an entirely new mission, with no training at all. Fine-tuning then becomes an optional refinement rather than a precondition for deployment.

Telecommands did not materially affect Chronos-2 (Table 1), even when revealed over the forecast horizon. ESA-ADB identifies commanded manoeuvres, resets and calibrations as rare nominal events and distinguishes commanded from anomalous resets [7]. Because our windows exclude every annotated event (Section 3), the transitions telecommands could explain are absent from training and test data.

The one-to-two-order-of-magnitude gap between mean and median MSE shows that a few difficult channels dominate the mean. We compute R^2 per channel relative to the training-mean predictor and aggregate across channels, omitting it when the baseline MSE is at most 10^{-8}; on near-constant `channel_69`, raw-unit MSE remains of order 10^{-10} even when R^2 is near or below zero. We therefore headline the median while reporting both aggregates and per-channel errors.

Limitations. All LSTM configurations share a fixed hidden size (Section 5), so the multivariate result characterises a fixed-capacity baseline rather than LSTMs in general. We evaluate one mission, point forecasts and a single LSTM baseline, leaving probabilistic detection and broader forecasting architectures for future work.

8 Conclusion

We compared an LSTM baseline against the foundation model Chronos-2 for forecasting spacecraft telemetry, with downstream anomaly detection as the goal. Chronos-2 is the stronger model in every configuration tested. It forecasts all 76 channels jointly and benefits from their interactions, whereas the LSTM degrades badly enough to be pushed towards one model per channel. Chronos-2's cumulative horizon-averaged error also grows slowly across the full forecast horizon. It achieves this in both zero-shot and fine-tuned settings, so a forecaster could be deployed on a new mission immediately, without first accumulating enough mission data to train on. The natural next step is to use the quantiles Chronos-2 already produces as the nominal envelope of a detector, replacing the residual thresholding stage that current methods depend on. Furthermore, evaluating the ability of Chronos-2 to leverage telecommand data in forecasting rare commanded events is crucial for anomaly detection.

Acknowledgements

We acknowledge the support of the European Space Agency (ESA), through the activity *AiSTRA-Nova: Knowledge Graph Reasoning for AI-Enhanced Spacecraft Health Monitoring*.

References

[1] Abdul Fatir Ansari et al. 2025. Chronos-2: From Univariate to Universal Forecasting. arXiv:2510.15821 [cs.LG]. (Oct. 2025). doi:10.48550/arXiv.2510.15821.

[2] Pawel Benecki, Szymon Piechaczek, Daniel Kostrzewa, and Jakub Nalepa. 2021. Detecting anomalies in spacecraft telemetry using evolutionary thresholding and LSTMs. *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 1143–1150. doi:10.1145/3449726.3459411.

[3] Edward J. Hu et al. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685 [cs.CL]. (Oct. 2021). doi:10.48550/arXiv.2106.09685.

[4] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. 2018. Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 387–395. doi:10.1145/3219819.3219845.

[5] Rob J. Hyndman and George Athanasopoulos. 2021. *Forecasting: Principles and Practice*. (3rd ed.). OTexts, Melbourne, Australia.

[6] Weihua Jin, Bo Sun, Zhidong Li, Shijie Zhang, and Zhonggui Chen. 2019. Detecting Anomalies of Satellite Power Subsystem via Stage-Training Denoising Autoencoders. *Sensors*, 19, 14, (July 22, 2019), 3216. doi:10.3390/s19143216.

[7] Krzysztof Kotowski et al. 2025. European Space Agency Benchmark for Anomaly Detection in Satellite Telemetry. arXiv:2406.17826 [cs.LG]. (Aug. 2025). doi:10.48550/arXiv.2406.17826.

[8] Siva Rama Krishna Kottapalli et al. 2025. Foundation Models for Time Series: A Survey. (Apr. 5, 2025). arXiv: 2504.04011 [cs.LG]. Retrieved Aug. 31, 2026 from http://arxiv.org/abs/2504.04011. Preprint.

[9] Daniel Lakey and Tim Schlippe. 2024. A Comparison of Deep Learning Architectures for Spacecraft Anomaly Detection. In *2024 IEEE Aerospace Conference*. (Mar. 2024), 1–11. doi:10.1109/AERO58975.2024.10521015.

[10] Daniel Lakey and Tim Schlippe. 2024. Anomaly Detection in Spacecraft Telemetry: Forecasting vs. Classification. In *2024 IEEE Space Computing Conference (SCC)*. IEEE, Mountain View, CA, USA, (July 2024), 17–27. doi:10.1109/SCC61854.2024.00009.

[11] Bryan Lim, Sercan Ö. Arık, Nicolas Loeff, and Tomas Pfister. 2021. Temporal Fusion Transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37, 4, (Oct. 2021), 1748–1764. doi:10.1016/j.ijforecast.2021.03.012.

[12] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. 2015. Long Short Term Memory Networks for Anomaly Detection in Time Series. *European Symposium on Artificial Neural Networks (ESANN)*, 89–94.

[13] Bogdan Ruszczak, Krzysztof Kotowski, David Evans, and Jakub Nalepa. 2025. The OPS-SAT benchmark for detecting anomalies in satellite telemetry. *Scientific Data*, 12, 1, (Apr. 2025), 710. doi:10.1038/s41597-025-05035-3.

[14] David Salinas, Valentin Flunkert, and Jan Gasthaus. 2019. DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks. arXiv:1704.04110 [cs.AI]. (Feb. 2019). doi:10.48550/arXiv.1704.04110.

[15] Renjie Wu and Eamonn J. Keogh. 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, (Mar. 2023), 2421–2429. doi:10.1109/TKDE.2021.3112126.

[16] Zhaoping Xu, Zhijun Cheng, and Bo Guo. 2023. A Multivariate Anomaly Detector for Satellite Telemetry Data Using Temporal Attention-Based LSTM Autoencoder. *IEEE Transactions on Instrumentation and Measurement*, 72, 1–13. doi:10.1109/TIM.2023.3296125.