IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151084

A Comparative Analysis of Prediction-Based and Reconstruction-Based Deep Learning Models for Time Series Anomaly Detection in Spacecraft Telemetry

Ashwin Krishna N[1], Dr.H.Mary Shyni[2]

M.Tech 2nd Year, Department of Computer Science and Engineering, SRMIST, Vadapalani[1]

Assistant Professor, Department of Computer Science and Engineering, SRMIST, Vadapalani[2]

Abstract: Anomaly detection in multivariate spacecraft telemetry is usually judged with the point-adjusted (PA) F1 score, a protocol known to reward any detector that fires at least once inside a labelled segment. This paper reports a controlled comparison of three deep models, an LSTM predictor, an LSTM autoencoder and a GRU autoencoder with additive attention, on twelve channels of the NASA SMAP and MSL benchmark. All models share one preprocessing, windowing, smoothing and thresholding pipeline and are scored with both PA and strict point-wise F1; each is also set against two trivial baselines, a random detector and the norm of the raw input. Under PA the random detector reaches a micro-averaged F1 of 0.96, above every trained model (0.79–0.81), while its point-wise F1 is only 0.02 against 0.44–0.45 for the networks. The raw-signal baseline matches the networks on both protocols (0.81 and 0.46). With a fixed threshold multiplier the autoencoders score zero PA F1 on 5 channels; once the multiplier is tuned per channel the three architectures become indistinguishable within run-to-run variation, although on three SMAP channels the PA scores of every model rest on only 2–14 correctly flagged points (point-wise F1 $\leq$ 0.03), and two contextual-anomaly channels remain undetected by all of them. The results indicate that evaluation protocol and threshold calibration, rather than architecture, account for most of the differences reported on this benchmark.

Keyword: Spacecraft telemetry, multivariate time series, LSTM, Anomaly detection, Autoencoder, attention, SMAP, MSL, point adjustment.

I. INTRODUCTION

Spacecraft report their health through large numbers of telemetry channels, and operators have traditionally watched them with fixed limit checks and expert review. Deep learning offers a way to learn normal behaviour from data and to flag departures from it, and two modelling ideas dominate the literature: a prediction-based detector forecasts the next telemetry values and treats a large forecast error as evidence of an anomaly, while a reconstruction-based detector compresses a window of telemetry and treats a poorly reconstructed window as abnormal [23].

The NASA Soil Moisture Active Passive (SMAP) and Mars Science Laboratory (MSL) telemetry released with LSTM-NDT [1] has become a standard public benchmark, and published models report F1 scores above 0.9 on it [2]–[5]. These numbers are, however, produced almost entirely with the point-adjustment protocol of [24], under which a whole labelled segment counts as detected as soon as one of its points is flagged. Several studies show that this protocol can reward detectors that carry little information, including random ones [6]–[10].

The closest prior work, Herrmann et al. [10], re-evaluated an LSTM, USAD and the Anomaly Transformer on SMAP, MSL and a private ESA data set. Their study did not include an attention-augmented reconstruction model and did not report results by anomaly type. Most other recent telemetry studies rely on private satellite data or injected anomalies [11]–[17], so their scores cannot be compared directly. This paper therefore asks a narrow question on the public benchmark: how much of the reconstruction-based and attention-based detectors survives a stricter protocol and a comparison with trivial baselines? The contributions are:

1) a controlled comparison of three architectures under one pipeline, scored with both PA and point-wise F1; 2) random and raw-signal baselines evaluated on the same channels with the same protocols;
3) an analysis of how the threshold multiplier changes the ranking of the models; and
4) a breakdown by anomaly type and spacecraft, together with an estimate of run-to-run variation.

II. RELATED WORK

A. Deep detectors for telemetry and multivariate series

 This work is licensed under a Creative Commons Attribution 4.0 International License

DOI: 10.17148/IJARCCE.2026.151084

LSTM-NDT [1] pairs an LSTM forecaster with non-parametric dynamic thresholding. OmniAnomaly [2] uses a stochastic recurrent network with a variational autoencoder, USAD [3] trains two autoencoders adversarially, MTAD-GAT [4] adds graph attention over features and time, and the Anomaly Transformer [5] scores an association discrepancy. Spacecraft-specific work has proposed a framework that filters transmission artefacts [13], a temporal-dependence Mahalanobis distance [14], reconstruction-discrepancy alerts [15], spatio-temporal graph learning [16], an architecture comparison [12] and edge-deployable models [17].

B. Critiques of evaluation

PA was introduced in [24] and is now the default on SMAP and MSL. Kim et al. [6] show that it can make random scores competitive, and Wu and Keogh [7] argue that common benchmarks are flawed. Garg et al. [8] and Sehili and Zhang [9] discuss metric choice and aggregation, Nalepa et al. [11] show that conclusions on satellite telemetry change with the metric, and Herrmann et al. [10] propose an event-based composite F score and add random, raw-signal and untrained-network baselines. The present work follows the same spirit but compares architecture families that [10] did not cover. Table I summarises the studies most relevant here.

TABLE I
SELECTED RELATED WORK

Ref.	Yea r	Focus	Relevance to this study
[1]	201 8	LSTM forecaster, dynamic threshold	Source of benchmark; prediction baseline
[2]	201 9	Stochastic RNN + VAE	Reconstruction family
[3]	202 0	Adversarial autoencoders	Reconstruction; re-tested in [10]
[4]	202 0	Graph attention, forecast + recon.	Attention family
[5]	202 2	Association discrepancy	Transformer; weak in [10]
[6]	202 2	Critique of PA	Random scores competitive under PA
Ref.	**Yea r**	**Focus**	**Relevance to this study**
[10]	202 4	Re-evaluation on SMAP/MSL	Base paper; baselines, event metrics
[11]	202 2	Satellite telemetry metrics	Metric choice alters conclusions
[12]	202 4	CNN/RNN/Transforme r comparison	Architecture comparison, other data
[15]	202 4	Reconstruction discrepancy	GEO attitude data, not public
[16]	202 5	Dynamic graph + GRU-attention VAE	Private/other data; protocol differs

III. DATASET AND PROBLEM SETTING

The benchmark [1] contains 82 channels, 55 from SMAP and 27 from MSL. Each channel is a multivariate stream of 25 (SMAP) or 55 (MSL) inputs made of one real-valued telemetry signal and binary command indicators. Training sequences contain no labelled anomalies. Test sequences carry labelled anomaly intervals, each tagged as a point or a contextual anomaly. A model is trained per channel on the training sequence and must flag the labelled intervals in the test sequence without using labels.

Twelve channels were used: eight from SMAP and four from MSL, six with point anomalies and six with contextual anomalies (Table II). They were chosen by hand to cover both spacecraft and both anomaly types, not sampled at random,

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151084

which limits how far the results generalise (Section VI). Formally, a window of the last W = 100 telemetry vectors is mapped to an anomaly score s(t), and an alarm is raised when s(t) exceeds a threshold.

TABLE II
CHANNELS USED IN THE EXPERIMENTS

Channel	Craft	Type	d	Test length	Anomalous points (%)
P-1	SMAP	contextual	25	8505	751 (8.8)
E-1	SMAP	contextual	25	8516	508 (6.0)
A-2	SMAP	contextual	25	7914	111 (1.4)
G-1	SMAP	contextual	25	8469	121 (1.4)
S-1	SMAP	point	25	7331	448 (6.1)
E-2	SMAP	point	25	8532	1398 (16.4)
D-1	SMAP	point	25	8509	3259 (38.3)
P-2	SMAP	point	25	8209	1226 (14.9)
M-1	MSL	contextual	55	2277	1141 (50.1)
M-2	MSL	contextual	55	2277	1141 (50.1)
M-6	MSL	point	55	2049	181 (8.8)
T-4	MSL	point	55	2217	69 (3.1)

IV. METHODOLOGY

A. Preprocessing

Each channel is standardized with the mean and standard deviation of its training sequence, and overlapping windows of length W = 100 with stride one are cut from the training and test sequences. Constant input columns are kept in the main experiments; in an earlier trial, choosing per channel between cleaned and uncleaned input changed micro-averaged F1 by at most 0.008.

B. Models

Three architectures are compared (Fig. 2, Table III). All use hidden size 32 and a mean-squared-error training loss on normal data.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151084

Fig. 1 shows the processing chain, which is identical for every model and channel.

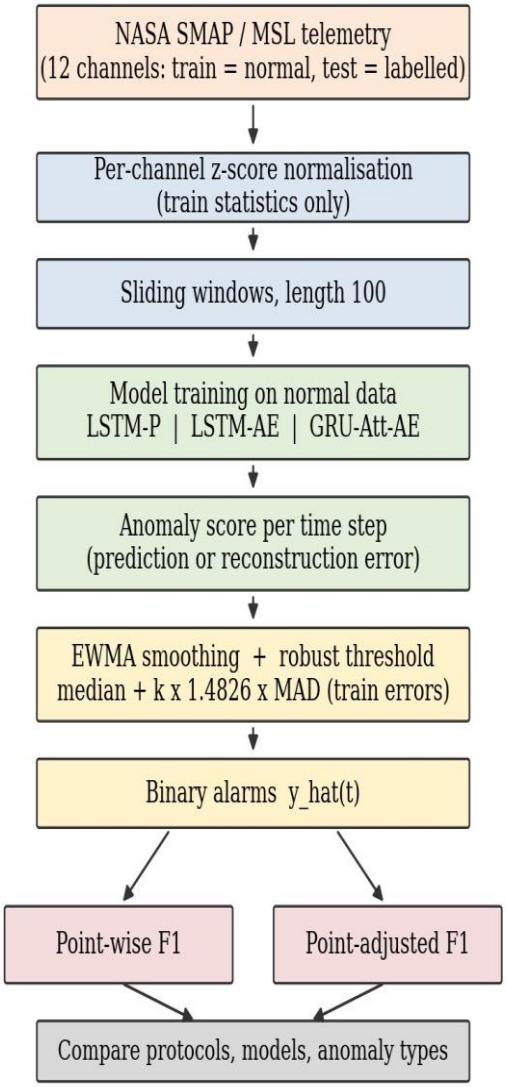


Fig. 1. Processing pipeline shared by all models.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151084

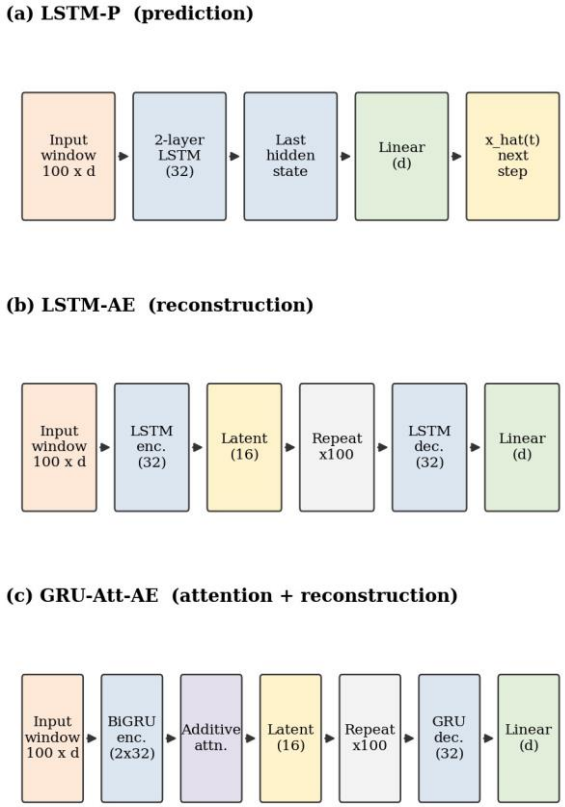


Fig. 2. Block diagrams of the three architectures (d is the number of input features).

LSTM-P is a two-layer LSTM [18] whose last hidden state is projected to a forecast of the next telemetry vector. LSTM-AE encodes the window with an LSTM, compresses the final state to a 16-dimensional latent code, repeats the code over the window length and decodes it with a second LSTM. GRU-Att-AE replaces the encoder with a bidirectional GRU [19], pools its outputs with additive attention [20], and decodes with a GRU. The models are deliberately small so that differences come from the modelling idea rather than from capacity.

TABLE III
MODEL SUMMARY

Model	Paradigm	Params (d=25)	Anomaly score
LSTM-P	Prediction	16,825	Squared forecast error of next step
LSTM-AE	Reconstructio n	17,897	Squared error at last step of window
GRU-Att-A E	Attention + recon.	24,297	Squared error at last step of window

Training uses Adam [21] with learning rate 10^{-3}, batch size 64 and 12 epochs, implemented in PyTorch [22] on a CPU. No hyper-parameter search was performed.

C. Scoring and thresholding

For the predictor the raw score is the mean squared error over the d features, and for the autoencoders it is the same quantity at the last time step of each window:

$e(t) = (1/d)\ \Sigma\ (\ x\ (t) - x'\ (t)\ )^2 (1)$

The score is smoothed with an exponentially weighted moving average and compared with a robust threshold computed from the smoothed training errors:

$s(t) = \alpha \cdot e(t) + (1 - \alpha) \cdot s(t - 1),\ \alpha = 2/(L + 1),\ L = 30 (2)\ \tau = \text{median}(s_train) + k \cdot 1.4826 \cdot \text{MAD}(s_train) (3)$

DOI: 10.17148/IJARCCE.2026.151084

An alarm is raised at time t when $s(t) > \tau$. The median and MAD make the threshold insensitive to the few large training errors that inflate a mean-plus-standard-deviation rule. The multiplier k is searched over {1.5, 2.0, …, 6.0}. Two settings are reported: a fixed k = 4, which uses no labels, and a best-k setting in which k is chosen per channel and model to maximise PA F1 on the labelled test sequence. The latter is an oracle in the spirit of the best-F1 protocol of [10]; it uses test labels, gives every model the same advantage, and should be read as an optimistic upper bound.

D. Evaluation protocols and baselines

Point-wise F1 compares every alarm with the label at the same time step. PA F1 first expands the alarms: if any point of a labelled segment is flagged, the whole segment is counted as detected [24]. F1 values are aggregated in two ways, micro (true and false positives and negatives pooled over the channels) and mean (average of per-channel F1). Pooling lets long anomalies dominate under PA, which [10] cautions against, so both are reported.

Two baselines use the same smoothing, threshold rule and best-k search. The random detector flags each time step independently with probability 0.01 and is averaged over 200 draws. The raw-norm detector uses the Euclidean norm of the standardised input vector as its score, with no learning.

E. Protocol for repeated runs

Each model-channel pair is trained once with seed 42. To gauge run-to-run variation, results are compared with an earlier independent run of the same pipeline under a different random state (Section V-D).

V. RESULTS AND DISCUSSION

A. Overall comparison under two protocols

Table IV and Fig. 3 give the main results. Under PA, the random detector attains a micro-averaged F1 of 0.96 and a mean per-channel F1 of 0.82, higher than any trained model (micro 0.79–0.81, mean 0.71–0.72). Its point-wise F1 collapses to 0.02, whereas the networks obtain 0.44–0.45. The protocol therefore separates informative from uninformative detectors only when point-wise scoring is used.

The raw-norm baseline, which learns nothing, reaches micro PA 0.81 and point-wise 0.46, indistinguishable from the three networks. On this subset the deep models therefore show no measurable benefit over a simple signal-magnitude rule, in line with the raw-signal behaviour reported in [10].

TABLE IV
F1 OVER 12 CHANNELS (PA = POINT-ADJUSTED, PW = POINT-WISE)

Method	Micro PA	Mean PA	Micro PW	Mean PW
Random (p = 0.01)	0.957	0.822	0.019	0.017
Raw norm (best k)	0.806	0.704	0.461	0.265
LSTM-P, k = 4	0.822	0.688	0.472	0.251
Method	**Micro PA**	**Mean PA**	**Micro PW**	**Mean PW**
LSTM-AE, k = 4	0.718	0.481	0.504	0.250
GRU-Att-AE, k = 4	0.719	0.481	0.503	0.247
LSTM-P, best k	0.794	0.715	0.442	0.246
LSTM-AE, best k	0.805	0.711	0.447	0.249
GRU-Att-AE, best k	0.805	0.712	0.447	0.248

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151084

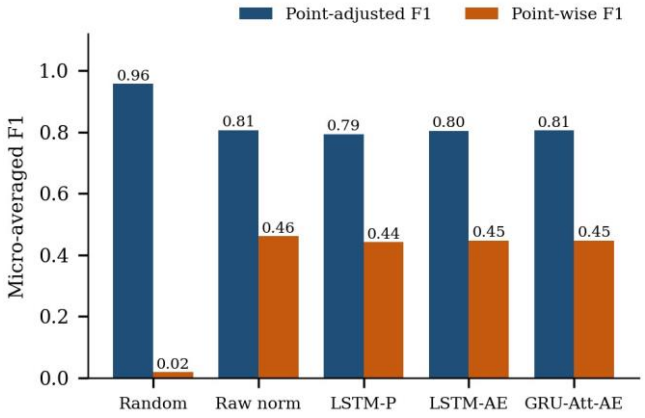


Fig. 3. Micro-averaged PA and point-wise F1 of the baselines and the three models (best-k setting).

B. Effect of the threshold multiplier

With the fixed k = 4 the architectures look very different: the autoencoders return zero PA F1 on 5 channels (E-1, A-2, G-1, S-1, E-2) and the predictor on 2 (A-2, G-1), giving mean PA F1 of 0.48 (LSTM-AE) and 0.48 (GRU-Att-AE) against 0.69 for the predictor. Fig. 4 shows that this ranking is not stable across a common multiplier: the predictor leads for 2.0 ≤ k ≤ 5.0 (by 0.21 at k = 4), the autoencoders lead at k = 1.5, and all three are level (within 0.01) for k ≥ 5.5. When k is tuned per channel, the mean PA F1 of the three models lies within 0.004 of each other (Table IV).

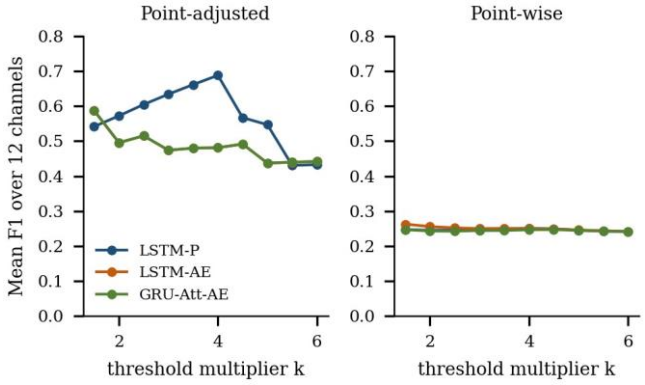


Fig. 4. Mean F1 over the 12 channels at a common threshold multiplier k.

The recovery of the autoencoders at tuned k deserves caution. Fig. 5 shows channel E-2, where the smoothed score of LSTM-AE inside the labelled anomaly is lower, not higher, than during normal operation (mean 0.06 against 0.45). At k = 1.5 only 4 of the 1,398 anomalous points exceed the threshold, yet PA turns them into a segment-level detection with F1 of 0.85, while the point-wise F1 is 0.00. The same pattern occurs on E-1 and S-1: at the best k, all three models flag only 2–14 anomalous points per channel on E-1, S-1 and E-2 (point-wise F1 ≤ 0.03) but obtain PA F1 between 0.68 and 0.88.

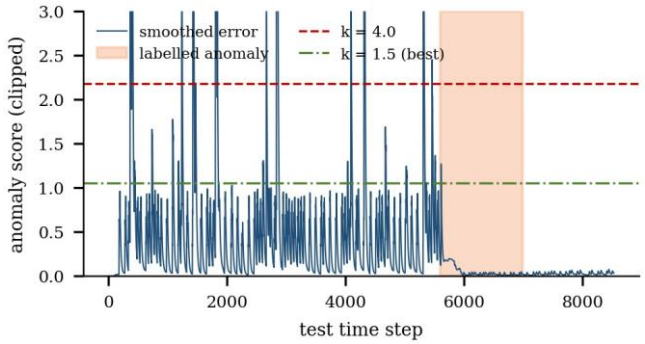


Fig. 5. Smoothed LSTM-AE score on channel E-2 (y-axis clipped; spikes reach 16). The score inside the labelled anomaly stays low; only 4 of its 1,398 points exceed the k = 1.5 threshold.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151084

The chosen k is often at the edge of the searched range: for the autoencoders it equals 6.0 on 5 channels and 1.5 on 4 (predictor: 5 and 2), so the true optimum may lie outside the grid. Because the best k depends on the model, a single fixed multiplier is not a neutral choice when architectures are compared.

C. Channels and anomaly types

Table V lists per-channel results and Fig. 6 shows them as a heat map. Point anomalies are detected far better than contextual ones: mean PA F1 is 0.86 for point channels and 0.56–0.57 for contextual channels, and the point-wise gap is larger still (0.45 against 0.04 for LSTM-P; Table VI). Channels A-2 and G-1, both contextual and short (111 and 121 anomalous points), are never detected by any model at any k in the grid. Conversely, the high PA scores on the SMAP channels E-1, S-1 and E-2 are not evidence of good localisation, since they rest on 2–14 correct points each (Section V-B).

TABLE V
PER-CHANNEL F1 AT BEST K (PA / PW)

Channel	Type	LSTM-P	LSTM-AE	GRU-Att-AE
P-1	C	0.74 / 0.07	0.74 / 0.07	0.74 / 0.07
E-1	C	0.75 / 0.01	0.68 / 0.03	0.68 / 0.02
A-2	C	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
G-1	C	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
S-1	P	0.77 / 0.01	0.76 / 0.01	0.76 / 0.01
E-2	P	0.88 / 0.01	0.85 / 0.00	0.85 / 0.01
D-1	P	0.97 / 0.92	0.98 / 0.92	0.98 / 0.93
P-2	P	0.91 / 0.85	0.94 / 0.88	0.94 / 0.88
M-1	C	0.97 / 0.08	0.98 / 0.06	0.98 / 0.06
M-2	C	0.97 / 0.09	0.98 / 0.10	0.98 / 0.10
M-6	P	0.94 / 0.90	0.94 / 0.90	0.94 / 0.90
T-4	P	0.67 / 0.01	0.69 / 0.02	0.69 / 0.02

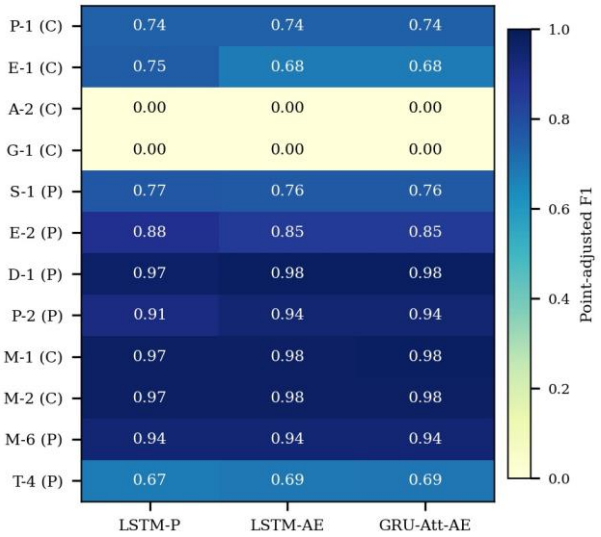


Fig. 6. PA F1 per channel and model at best k (P = point, C = contextual anomaly channel).

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151084

TABLE VI
MEAN F1 BY ANOMALY TYPE AND SPACECRAFT (PA / PW)

Group	n	LSTM-P	LSTM-AE	GRU-Att-AE
Point	6	0.86 / 0.45	0.86 / 0.45	0.86 / 0.46
Contextual	6	0.57 / 0.04	0.56 / 0.04	0.56 / 0.04
SMAP	8	0.63 / 0.23	0.62 / 0.24	0.62 / 0.24
MSL	4	0.89 / 0.27	0.89 / 0.27	0.90 / 0.27

By spacecraft, the models score 0.62–0.63 (mean PA) on the eight SMAP channels and 0.89–0.90 on the four MSL channels. On the SMAP channels the random detector scores 0.83 and the raw-norm detector 0.60, so the trained models are below the former and level with the latter; on MSL the random and raw-norm baselines score 0.82 and 0.92. Channels M-1 and M-2 have identical test length and anomaly extent and are therefore not independent samples.

D. Run-to-run variation and ablations

Table VII compares the micro PA F1 obtained with k = 4 for window lengths of 50, 100 and 150. The window length was varied in an earlier run (run A); the main run of this paper is run B. For W = 100 the two runs differ by 0.039 for the predictor, more than the largest effect of changing the window (0.024), and by under 0.001 for the autoencoders. At best k, run A gave micro PA F1 of 0.768, 0.803 and 0.808 for the three models, against 0.794, 0.805 and 0.805 here. Differences of this size between the two autoencoders, or between any of the three models at best k, should therefore not be interpreted as a ranking. A single main run per configuration is a limitation, and multiple seeds are planned.

TABLE VII
MICRO PA F1 AT K = 4 FOR DIFFERENT WINDOW LENGTHS

Model	W=50 (A)	W=100 (A)	W=100 (B)	W=150 (A)
LSTM-P	0.759	0.783	0.822	0.780
LSTM-AE	0.707	0.718	0.718	0.717
GRU-Att-AE	0.713	0.719	0.719	0.718

E. Relation to the base study

Table VIII places the per-dataset mean PA F1 of this study next to the PA F1 reported in Table 6 of [10]. The comparison is indicative only. Herrmann et al. use all 69 SMAP and 36 MSL anomaly sequences, whereas this study uses eight and four channels; their scores are averaged over sequences, ours over channels; and our best-k thresholds use the test labels. The qualitative picture agrees: under PA a random detector scores highly and the raw signal is hard to beat. In numbers, the LSTM predictor here is below the LSTM with non-parametric thresholding of [10] on SMAP channels (0.63 against 0.79) and above it on MSL channels (0.89 against 0.64), so no consistent advantage over the base study is claimed.

TABLE VIII
MEAN PA F1 PER DATASET: THIS STUDY VS. [10]

Source	Method	SMAP	MSL
This study	Random	0.83	0.82
This study	Raw norm	0.60	0.92
This study	LSTM-P (best k)	0.63	0.89
This study	LSTM-AE (best k)	0.62	0.89
This study	GRU-Att-AE (best k)	0.62	0.90
[10]	Random	0.64	0.69
[10]	Raw signal	0.50	0.48
[10]	LSTM (best-F1 threshold)	0.61	0.45

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151084

[10]	LSTM (NPT)	0.79	0.64
[10]	USAD	0.52	0.68
[10]	Anomaly Transformer	0.60	0.57

F. Discussion

Three practical points follow. First, PA F1 on SMAP and MSL should never be reported without a random and a raw-signal baseline, since both can match or exceed trained models. Second, point-wise F1 exposes differences that PA hides, although it can underrate detectors when labels are coarse [6]; event-based metrics such as the composite F score of [10] are a sensible complement. Third, the threshold rule is part of the model: with a fixed multiplier the reconstruction models appear to fail, and with a tuned one they match the predictor, but on several channels that match consists of a few isolated alarms credited to whole segments by PA. Comparisons of architectures therefore need a stated common threshold policy and point-wise or event-level reporting.

VI. LIMITATIONS

The study covers 12 of 82 channels chosen by hand, and two of them are near-duplicates. Best-k thresholds and the channel-wise threshold selection use test labels and are optimistic; for many channels the selected k lies at the boundary of the grid. Each configuration was trained once in the main run, and the observed run-to-run differences are of the same order as the gaps between models. The models are small, trained for 12 epochs without tuning, and no event-based metric or graph or Transformer model was evaluated. The comparison with [10] is indirect for the reasons given in Section V-E.

VII. CONCLUSION AND FUTURE WORK

On twelve SMAP and MSL channels, three deep detectors of different design were indistinguishable once their thresholds were calibrated, a random detector outscored all of them under point adjustment, and a learning-free signal-norm baseline matched them under both protocols. On three SMAP channels, the point-adjusted scores of every model came from a handful of correctly flagged points. Contextual anomalies, especially short ones, were not detected by any model. These findings suggest that reported differences on this benchmark depend more on the evaluation protocol and threshold policy than on architecture.

Future work will extend the study to all 82 channels with several random seeds, add event-based metrics such as the composite F score and PA%K [6], [10], include a Transformer and a graph-based detector, replace label-tuned thresholds with label-free rules such as dynamic or extreme-value thresholds, and analyse why contextual anomalies escape both prediction and reconstruction scores.

REFERENCES

[1] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, “Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding,” in Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2018, pp. 387–395.

[2] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, “Robust anomaly detection for multivariate time series through stochastic recurrent neural network,” in Proc. 25th ACM SIGKDD, 2019, pp. 2828–2837.

[3] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, “USAD: Unsupervised anomaly detection on multivariate time series,” in Proc. 26th ACM SIGKDD, 2020, pp. 3395–3404.

[4] H. Zhao et al., “Multivariate time-series anomaly detection via graph attention network,” in Proc. IEEE ICDM, 2020, pp. 841–850.

[5] J. Xu, H. Wu, J. Wang, and M. Long, “Anomaly Transformer: Time series anomaly detection with association discrepancy,” in Proc. ICLR, 2022 (arXiv:2110.02642).

[6] S. Kim, K. Choi, H.-S. Choi, B. Lee, and S. Yoon, “Towards a rigorous evaluation of time-series anomaly detection,” in Proc. AAAI Conf. Artif. Intell., vol. 36, 2022, pp. 7194–7201.

[7] R. Wu and E. Keogh, “Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress,” IEEE Trans. Knowl. Data Eng., 2021, doi: 10.1109/TKDE.2021.3112126.

[8] A. Garg, W. Zhang, J. Samaran, R. Savitha, and C.-S. Foo, “An evaluation of anomaly detection and diagnosis in multivariate time series,” IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 6, pp. 2508–2517, 2021.

[9] M. El Amine Sehili and Z. Zhang, “Multivariate time series anomaly detection: Fancy algorithms and flawed evaluation methodology,” arXiv e-prints, Aug. 2023.

[10] L. Herrmann, M. Bieber, W. J. C. Verhagen, F. Cosson, and B. F. Santos, “Unmasking overestimation: a re-evaluation of deep anomaly detection in spacecraft telemetry,” CEAS

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151084

Space J., vol. 16, pp. 225–237, 2024, doi: 10.1007/s12567-023-00529-5.

[11] J. Nalepa et al., "Evaluating algorithms for anomaly detection in satellite telemetry data," Acta Astronaut., vol. 198, pp. 689–701, 2022, doi: 10.1016/j.actaastro.2022.06.026.

[12] D. Lakey and T. Schlippe, "A comparison of deep learning architectures for spacecraft anomaly detection," in Proc. IEEE Aerosp. Conf., 2024, doi: 10.1109/AERO58975.2024.10521015.

[13] Y. Wang, J. Gong, J. Zhang, and X. Han, "A deep learning anomaly detection framework for satellite telemetry with fake anomalies," Int. J. Aerosp. Eng., 2022, doi: 10.1155/2022/1676933.

[14] J. Pang, D. Liu, Y. Peng, and X. Peng, "Temporal dependence Mahalanobis distance for anomaly detection in multivariate spacecraft telemetry series," ISA Trans., vol. 140, pp. 354–367, 2023, doi: 10.1016/j.isatra.2023.06.002.

[15] M. Liu, Q. Xia, and S. Qiu, "A new data-driven framework for progressive anomaly event alerts in spacecraft based on reconstruction discrepancy," Adv. Space Res., vol. 74, no. 11, pp. 5890–5905, 2024, doi: 10.1016/j.asr.2024.08.054.

[16] Y. Lai et al., "STGLR: A spacecraft anomaly detection method based on spatio-temporal graph learning," Sensors, vol. 25, no. 2, p. 310, 2025, doi: 10.3390/s25020310.

[17] C. Goetze, T. Schlippe, and D. Lakey, "Deep learning-based anomaly detection in spacecraft telemetry on edge devices," in Proc. IEEE Space Comput. Conf., 2025, doi: 10.1109/SCC66396.2025.00019.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[19] K. Cho et al., "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in Proc. EMNLP, 2014, pp. 1724–1734.

[20] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in Proc. ICLR, 2015.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. ICLR, 2015.

[22] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in Adv. Neural Inf. Process. Syst., vol. 32, 2019.

[23] G. Pang, C. Shen, L. Cao, and A. van den Hengel, "Deep learning for anomaly detection: A review," ACM Comput. Surv., vol. 54, no. 2, pp. 1–38, 2021.

[24] H. Xu et al., "Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications," in Proc. World Wide Web Conf., 2018, pp. 187–196.

 This work is licensed under a Creative Commons Attribution 4.0 International License