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 11, November 2026

Unsupervised Anomaly Detection for Technosignature Discovery in Breakthrough Listen Radio Archives

Aishwarya K[1], Poornima S[2]

Department of Computer Science (Emerging Technologies), SRMIST, Vadapalani Chennai, India[1]

Department of Computer Science (Emerging Technologies), SRMIST, Vadapalani Chennai, India[2]

Abstract: Radio searches for technosignatures must sift petabyte-scale archives dominated by noise and interference, and rule-based Doppler-drift detectors such as turboSETI can only find signals that match their search model. This paper develops and benchmarks a signal-model-free alternative: a convolutional autoencoder (CAE) trained only on background data, whose reconstruction error serves as an anomaly score. A reproducible framework evaluates the CAE against turboSETI and non-learned baselines, using setigen signal injection at matched false-alarm rates and 1,200 held-out cadences of the Kaggle SETI Breakthrough Listen E.T. Signal Search data, with ROC-AUC, average precision, precision, recall and F1. The study shows that turboSETI is the more sensitive detector for linear-drift signals at matched false-alarm rates, whereas the CAE needs no drift model; that learned scores perform comparably to hand-designed ON–OFF statistics, at modest absolute levels in the tested configuration; and, through an anomaly taxonomy, that the strongest reconstruction errors fall on interference-like structure. Together these results provide reference points for learned technosignature detectors and identify drift-aware and ON–OFF-aware extensions as the next step.

Keywords: SETI, technosignatures, anomaly detection, convolutional autoencoder, Breakthrough Listen, turboSETI, setigen, radio-frequency interference

I. INTRODUCTION

Whether technology exists elsewhere in the Universe is among the oldest open questions in science, and radio astronomy offers one of its most practical probes. Since Cocconi and Morrison showed that interstellar communication could be sought at radio wavelengths [1], narrowband signals have been regarded as a leading technosignature, because narrow spectral lines are not known to arise from natural astrophysical processes. Modern searches are limited less by telescope time than by data: Breakthrough Listen publicly archives observations at petabyte scale [2], and every spectrogram mixes receiver noise, natural sources and radio-frequency interference (RFI) from human technology.

The workhorse search, turboSETI [3], looks for narrowband hits whose frequency drifts linearly with time, as expected from the relative acceleration of transmitter and receiver. It is fast and well understood, but it finds only what its search model describes. Machine learning offers a complementary route. Supervised approaches classify known signal types [4], yet confirmed extraterrestrial examples do not exist, so recent work has turned to anomaly detection, which learns what ordinary data look like and flags departures from it [5], [6], [7].

The problem therefore has three parts. Signals of interest are rare and may be weak, hidden among noise and interference in very large observations. Search algorithms find only the patterns their parameters describe, so unusual signals may pass unnoticed. And because no confirmed extraterrestrial signals exist, supervised models cannot be trained on genuine examples.

Reconstruction-error anomaly detection with autoencoders addresses the last of these directly, but its sensitivity relative to classical searches has rarely been measured under matched conditions. This paper builds that measurement. We train a convolutional autoencoder (CAE) on background data only and compare it with turboSETI and with simple non-learned statistics, holding every method to the same false-alarm rate and scoring it against signals of known strength. The contributions are:

- a CAE anomaly-scoring pipeline for single radio frames (1024 × 16) and six-snippet ON–OFF cadences (6 × 272 × 256), trained without any example of the target signal;
- a matched-false-alarm-rate comparison with turboSETI on setigen-injected signals, reported with Wilson confidence intervals;

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

- a full classification evaluation on 1,200 held-out Kaggle cadences (ROC-AUC, average precision, precision, recall, F1, specificity and balanced accuracy), with two non-learned baselines and an anomaly taxonomy; and
- design implications for drift-aware, cadence-aware learned detectors, with a reproducible single-script implementation.

The study is deliberately scoped. It uses selected Kaggle training samples and generated frames rather than the whole Breakthrough Listen archive; the Kaggle labels mark simulated signals, not confirmed transmissions; and normal examples are selected with the labels, so the cadence experiment is a label-assisted, normal-only form of anomaly detection. Within that scope the work offers a measured comparison of learned and classical detection, a way to inspect what a learned detector responds to, and a foundation that extends to larger datasets and stronger architectures.

II. RELATED WORK

Learned detectors have entered technosignature searches along several lines. Pardo et al. [5] apply machine-learning anomaly detection to Breakthrough Listen observations and rank signals by their frequency and time behaviour, and Ma et al. [6] search 820 nearby stars with a β-convolutional variational autoencoder, cutting the number of candidates requiring inspection. Painter et al. [8] combine statistical and RFI-vetting steps to search a very large Green Bank Telescope archive. Earlier, Harp et al. [4] applied machine vision and deep learning to the classification of radio SETI signals.

Beyond SETI, CARA [7] compares convolutional autoencoders with conventional models for detecting radio anomalies, and Li et al. [9] use an improved U-Net with semi-supervised learning to detect RFI.

Evaluation is a recurring difficulty because real positives do not exist. Setigen [10] addresses it by injecting physically motivated synthetic signals into data, and the analysis of the BLc1 candidate [11] illustrates why any candidate must be verified against interference. The present study uses both ideas: it evaluates the learned and classical detectors on the same injected frames at a matched false-alarm rate, and complements this with a labelled cadence benchmark.
Table I summarises the closest related studies, and Table II compares the main families of technique.

TABLE I. SUMMARY OF RELATED STUDIES

Ref.	Focus	Key contribution
[5]	Anomaly detection on Breakthrough Listen data	Ranks unusual signals by frequency and time behaviour
[6]	β-VAE search of 820 nearby stars	Reduces the number of candidates needing inspection
[8]	Green Bank Telescope archive	Statistical and RFI-vetting search of a very large archive
[10]	Synthetic signal injection	Open framework for physically motivated technosignatures
[11]	Verification of candidate BLc1	Separates extraterrestrial from interference origins
[7]	Autoencoders for radio anomalies	Reconstruction error useful for anomaly identification
[9]	Semi-supervised RFI detection	Improved U-Net identifies and removes interference
[1]	Interstellar communication	Founding proposal for radio searches
[2]	Breakthrough Listen data products	Public data, formats, reduction and archiving
[4]	Radio SETI signal classification	Machine vision and deep learning classifiers

TABLE II. COMPARISON OF DETECTION APPROACHES

Approach	Strength	Limitation
Traditional detection	Effective for expected narrowband and drift patterns	Misses signals outside predefined criteria
turboSETI	Scans large datasets for drifting narrowband signals	Depends on SNR threshold and drift limits
Supervised learning	Learns signal versus non-signal from labels	Needs labelled examples; none are confirmed
RFI detection	Identifies human-made interference	Interference and candidates can overlap
Anomaly detection	Learns complex patterns; error flags unusual data	Noise and RFI can also score high

The gap this work addresses follows from the comparison: classical searches are tied to predefined signal patterns, supervised learning needs examples that do not exist, and unusual signals may arise from noise or interference rather than from extraterrestrial sources. We therefore study convolutional autoencoders for anomaly detection on Kaggle Breakthrough Listen cadences and on controlled synthetic signals, evaluating the first against simulated-signal labels and the second against turboSETI.

III. FRAMEWORK OVERVIEW

Fig. 1 summarises the framework. The detection pipeline (left) converts raw filterbank data into spectrograms, scores each window by CAE reconstruction error, applies an ON–OFF cadence filter that removes locally generated interference, and outputs a ranked candidate list. The evaluation branch (right) exercises the same scoring on labelled Kaggle cadences, where classification metrics are available, and on setigen injections, where signal strength is known and turboSETI can be run on identical frames.

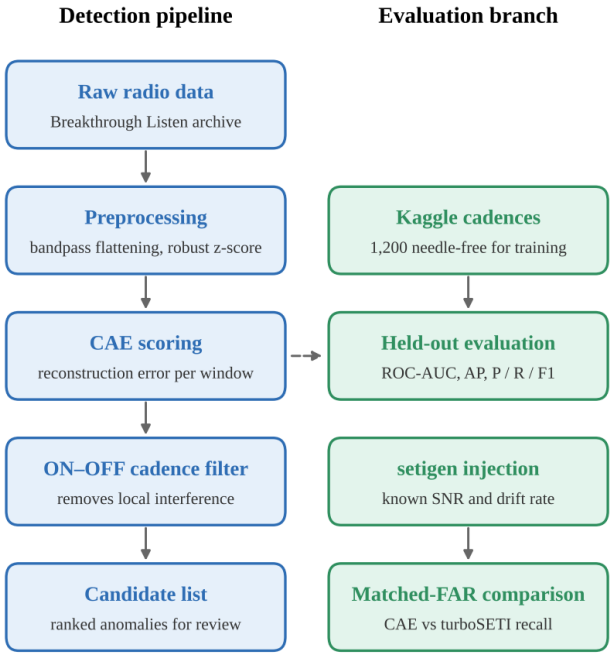


Fig. 1. Framework overview: detection pipeline (left) and evaluation branch (right).

Fig. 2.

The implementation is organised into five modules. The data preparation module loads the Kaggle cadences and labels, generates synthetic frames and prepares model inputs. The autoencoder module holds the single-frame and cadence CAEs. The anomaly detection module turns reconstruction error into scores. The evaluation module computes ROC-AUC, average precision, precision, recall and F1 for the Kaggle experiment and recall at a controlled false-alarm rate, with the turboSETI comparison, for the injection study. The visualisation module produces recall curves, error maps, ROC curves and taxonomy plots.

IV. DATA AND PREPROCESSING

A. Synthetic Frames

Frames of 1024 frequency channels × 16 time bins are generated with setigen [10] using the geometry of Breakthrough Listen high-frequency-resolution products (Δf = 2.794 Hz, Δt = 18.25 s, about 292 s per scan, starting at 6000 MHz). Background noise is chi-squared distributed (mean 10, standard deviation 5). Injected signals are Gaussian-profile narrowband lines (10 Hz width) with a constant drift rate whose magnitude is drawn uniformly from 0.1 to 0.8 Hz/s with random sign. Stationary RFI-like lines (SNR 4 to 25, zero drift) are added to half of the training frames.

B. Kaggle Cadence Data

The Kaggle SETI Breakthrough Listen – E.T. Signal Search data [12] store each observation as six 273 × 256 snippets in ON–OFF–ON–OFF–ON–OFF order. A binary label marks cadences into which a narrowband “needle” was injected in the ON snippets; the labels mark simulated signals, not extraterrestrial transmissions. The training set provides 60,000

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

labelled cadences, of which 6,000 (10%) contain a needle. Files are shuffled with a fixed seed and 3,000 are used: 60% form a training pool and 40% (1,200 cadences) form the held-out set, which contains 109 needles (9.1%). Time rows are cropped from 273 to 272.

C. *Preprocessing*

For synthetic frames, the time-averaged spectrum is divided by a 257-channel median-filtered baseline to flatten the bandpass, and the frame is robustly standardised with the median and the median absolute deviation (scaled by 1.4826). For cadence snippets, each snippet is bandpass-flattened with a 33-channel median filter (division for raw power, subtraction for already normalised data), robustly standardised and clipped to [−20, 20].

V. METHOD

A. *Autoencoder Architectures*

The single-frame CAE has four 3 × 3 convolutional layers (1→8→16→16→4 channels) with ReLU and max pooling, giving a 4 × 4 × 128 latent map (about 8× compression of the 16 × 1024 input). The decoder mirrors the encoder with convolutions and nearest-neighbour upsampling. The cadence CAE processes all six snippets as channels: two stride-2 convolutions (6→16→32) and a final convolution to 8 channels give an 8 × 68 × 64 latent map (about 12× compression), and two stride-2 transposed convolutions reconstruct the input. Both are trained to minimise the mean squared error (MSE) between input and reconstruction, and neither sees an example of the target signal during training.

B. *Anomaly Scores*

For a frame, the squared reconstruction error is averaged over time to give a per-channel profile, which is smoothed with a 9-channel moving average; the frame score is the maximum of this profile. For cadences, three scores are computed from the smoothed profile: error over all six snippets (AE-total), error over the ON snippets only (AE-ON), and error present in the ON but not the OFF snippets (AE-ON−OFF).

C. *Non-learned Baselines*

turboSETI [3] is run on each synthetic frame with a maximum drift of 1.0 Hz/s, a minimum drift of 0.05 Hz/s (near-stationary hits are treated as RFI) and a minimum SNR of 4. For cadences, two baselines use no learning: the ON-minus-OFF baseline is the maximum of the smoothed difference between the mean ON and mean OFF spectra, and the de-Doppler baseline applies a simple drift search (73 slopes up to ±0.4 channels per row) to the ON-minus-OFF image.

D. *Algorithm Summary*

The procedure is: (1) load the Kaggle cadences and labels and generate synthetic frames; (2) preprocess and arrange the data into model inputs; (3) select needle-free cadences for training; (4) train the CAE by minimising reconstruction MSE; (5) score held-out data by reconstruction error; (6) evaluate the cadence scores with ROC-AUC, average precision, precision, recall and F1; (7) inject signals at several SNR levels, set thresholds at a 5% false-alarm rate and measure recall; and (8) compare with turboSETI and analyse the results with plots and the anomaly taxonomy.

E. *Implementation*

The system is written in Python. NumPy and SciPy handle numerical processing and signal analysis, PyTorch builds and trains the autoencoders, scikit-learn provides ROC-AUC, average precision and k-means clustering, Matplotlib produces the figures, setigen generates and injects synthetic signals, and turboSETI serves as the classical baseline. The scanning module reads filterbank files with blimpy [13].

VI. EXPERIMENTAL PROTOCOL AND METRICS

Table III lists the configuration of both experiments; the whole study runs from one script, and the cadence model trained end to end in about 13 minutes in our environment.

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

TABLE III. EXPERIMENTAL CONFIGURATION

Setting	Injection study	Kaggle cadences
Input	1 × 16 × 1024 frame	6 × 272 × 256 cadence
Latent map	4 × 4 × 128	8 × 68 × 64
Training data	1,200 synthetic frames	1,200 needle-free cadences
Epochs / batch	14 / 32	8 / 16
Optimiser	Adam, 2×10^{-3}	Adam, 10^{-3}
Loss	MSE	MSE
Test data	8 SNR levels × 14 trials	1,200 cadences (109 needles)
Operating point	5% false alarms on 80 noise frames	nominal 5% false alarms

A. Synthetic Sensitivity Sweep

For each detector the threshold is the 95th percentile of its maximum score over 80 pure-noise frames, so each method raises a false alarm on 5% of noise frames. Signals are injected at SNR ∈ {2, 3, 4, 5, 6, 8, 10, 15}, with 14 trials per level. A detection counts only if the detector fires within the signal's true frequency span plus a 10-channel margin, so detections elsewhere in the frame earn no credit. Recall is reported with 95% Wilson score intervals [14].

B. Cadence Evaluation

The cadence CAE is trained on 1,200 needle-free cadences from the training pool with the Adam optimiser [15]. Labels are used only to select needle-free training cadences, so the experiment is label-assisted, normal-only anomaly detection rather than fully unsupervised. The training loss fell by 63% over eight epochs (3.48 to 1.27).

C. Evaluation Metrics

Two threshold-free metrics are reported: ROC-AUC and average precision (AP), the latter being more informative under class imbalance [16]; chance AP equals the prevalence (0.091). Threshold-based metrics are computed at a fixed operating point corresponding to a nominal 5% false-alarm rate on needle-free cadences. With TP, FP, FN and TN denoting true positives, false positives, false negatives and true negatives:

- Precision = TP / (TP + FP);
- Recall = TP / (TP + FN);
- F1 = 2 · Precision · Recall / (Precision + Recall);
- Specificity = TN / (TN + FP);
- Balanced accuracy = (Recall + Specificity) / 2;
- Accuracy = (TP + TN) / (TP + FP + FN + TN).

Accuracy is reported for completeness only: a classifier that never raises an alarm scores 0.909 on this held-out set, so accuracy cannot rank the methods. With 109 positives and 1,091 negatives, the Hanley–McNeil standard error [17] of an AUC near 0.55 is about 0.03.

VII. RESULTS

A. Injection Study: Learned versus Classical Detection

Table IV and Fig. 2 give recall at a matched 5% false-alarm rate. turboSETI's recall rises from 0.36 at SNR 4 to 0.93 at SNR 5 and reaches 1.00 from SNR 6. The CAE, which assumes nothing about drift rate, begins to recover signals at SNR 8 (0.07) and detects half of them at SNR 15. The gap quantifies the sensitivity cost of a signal-model-free, global-maximum score at a 5% false-alarm rate, and gives a concrete reference point for later architectures.

TABLE IV. RECALL OF INJECTED SIGNALS AT 5% FALSE-ALARM RATE

SNR	turboSETI	CAE
2	0/14 (0.00)	0/14 (0.00)
3	0/14 (0.00)	0/14 (0.00)
4	5/14 (0.36)	0/14 (0.00)
5	13/14 (0.93)	0/14 (0.00)
6	14/14 (1.00)	0/14 (0.00)
8	14/14 (1.00)	1/14 (0.07)
10	14/14 (1.00)	1/14 (0.07)
15	14/14 (1.00)	7/14 (0.50)

Counts correspond to the recall values in Fig. 2 (14 trials per SNR).

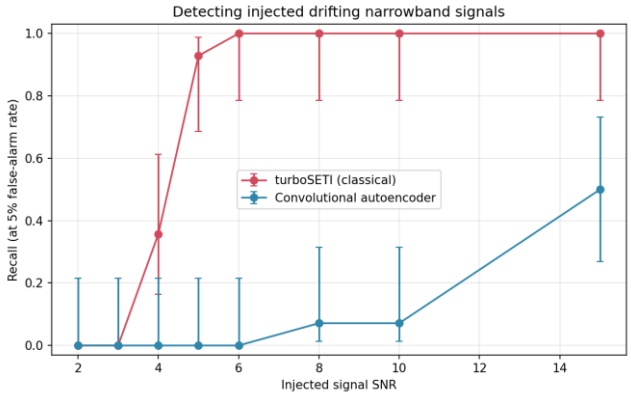


Fig. 3. Recall of injected drifting narrowband signals versus SNR at a 5% false-alarm rate, with 95% Wilson intervals.

Fig. 3 shows what limits the CAE. A faint drifting line adds little squared error compared with the error that ordinary noise fluctuations produce, so the frame's maximum error rarely rises above the 95th-percentile noise level until the line is bright.

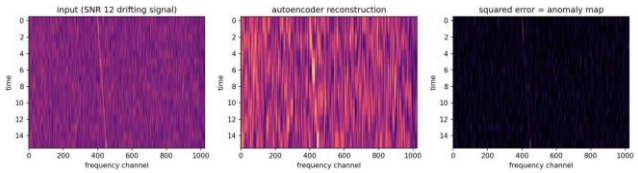


Fig. 4. Input frame with an SNR-12 drifting signal, the CAE reconstruction, and the squared-error anomaly map.

B. Kaggle Cadences: Threshold-free Metrics

Table V and Fig. 4 report ROC-AUC and AP on the 1,200 held-out cadences. All five scores lie close to chance in the tested configuration, with AUC between 0.527 and 0.556 and AP between 0.100 and 0.107 against a prevalence of 0.091. The learned scores match the hand-designed baselines (0.527–0.556 against 0.528–0.537) without any built-in notion of ON–OFF contrast or drift, and AE-ON−OFF is highest on both AUC and AP. As a sanity check on the evaluation code, the same pipeline run on generated cadences with strong planted needles reached ROC-AUC values of 0.92 (AE-total) and 1.00 (de-Doppler baseline).

TABLE V. ROC-AUC AND AVERAGE PRECISION ON HELD-OUT CADENCES

Score	ROC-AUC	Avg. precision
AE-total	0.527	0.100
AE-ON	0.541	0.103
AE-ON−OFF	0.556	0.107
ON−OFF baseline	0.528	0.101
De-Doppler baseline	0.537	0.100
Chance	0.500	0.091

n = 1,200 cadences, 109 with a needle (prevalence 0.091).

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

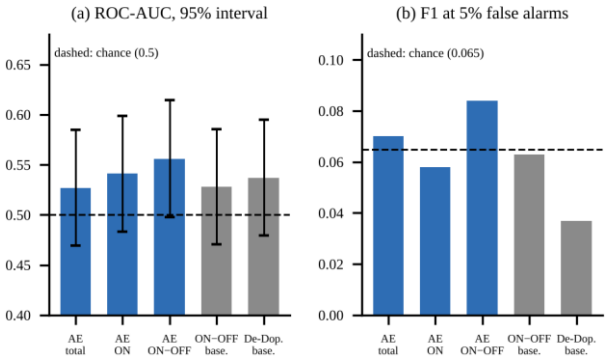


Fig. 5. Held-out cadence results: (a) ROC-AUC with 95% Hanley–McNeil intervals; (b) F1 at the 5% false-alarm operating point. Blue: autoencoder scores; grey: non-learned baselines.

C. *Kaggle Cadences: Operating-point Metrics*

Table VI gives precision, recall and F1 at the nominal 5% false-alarm operating point. Realised specificities are 0.946–0.960 (false-alarm rates of 4.0–5.4%). Precision ranges from 0.056 to 0.123, recall from 0.028 to 0.064 and F1 from 0.037 to 0.084, with AE-ON−OFF highest on all three. These values sit around the levels expected without skill (precision equal to the prevalence of 0.091, recall about 0.05 and F1 about 0.065). Balanced accuracy is 0.490–0.509, and accuracy (0.864–0.877) is below the 0.909 of a classifier that never fires.

TABLE VI. OPERATING-POINT METRICS AT 5% NOMINAL FALSE-ALARM RATE

Score	TP	FP	FN	TN	Prec.	Rec.	F1
AE-total	6	57	103	1034	0.095	0.055	0.070
AE-ON	5	59	104	1032	0.078	0.046	0.058
AE-ON−OFF	7	50	102	1041	0.123	0.064	0.084
ON−OFF base.	5	44	104	1047	0.102	0.046	0.063
De-Doppler base.	3	51	106	1040	0.056	0.028	0.037
Chance	—	—	—	—	0.091	0.050	0.065

Prec. = precision; Rec. = recall. Chance row: expected values at 5% false alarms with prevalence 0.091.

D. *Anomaly Taxonomy*

To understand what the CAE responds to, the top 15% of held-out cadences by AE-total (180 cadences) were clustered with k-means on four morphology features: the log ratio of ON to OFF error at the flagged location, the absolute drift slope, the width of the high-error region and the number of snippets showing elevated error. A heuristic rule labelled all four clusters RFI-like, because the flagged error appears in nearly all six snippets (Table VII). Roughly 13% of the flagged cadences (about 23 of 180) contained a needle, against 9.1% overall. The strongest reconstruction errors therefore fall mostly on persistent structure that is also present in the OFF snippets, which is the signature of interference rather than of an ON-only signal.

TABLE VII. CLUSTERS OF THE MOST STRONGLY FLAGGED CADENCES

Cluster	n	Needle %	ON/OFF	|Slope|	Snippets hit
0	35	6	0.49	0.011	6
1	134	15	0.02	0.000	6
2	5	20	−1.59	0.011	5
3	6	0	0.08	0.344	6

Needle % is the share of cadences in a cluster containing a needle somewhere; labels carry no location. ON/OFF is the median log error ratio; |Slope| is in channels per row.

E. *Scanning Module Demonstration*

The scanning module slides 1024-channel windows with 50% overlap across a filterbank or HDF5 file, resamples the time axis to 16 rows, scores each window with the single-frame CAE, ranks windows by peak error, clusters the latent codes of

the top windows with k-means, and annotates each hit with an approximate known-band label. In a demonstration on a sample file, the two highest-ranked windows (scores 7.30 and 6.58) were centred near 8419.386 MHz and show a strong narrowband line drifting across the frame (Fig. 5), while the third (1.66) lay close by at 8419.397 MHz (Table VIII). The built-in table, which is approximate and must be checked against official allocations before any claim, associates this region with deep-space X-band downlinks. This is a qualitative demonstration rather than a quantitative evaluation.

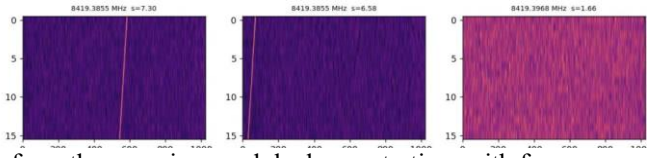


Fig. 6. Top-ranked windows from the scanning-module demonstration, with frequency and anomaly score above each panel.

TABLE VIII. TOP-RANKED WINDOWS IN THE DEMONSTRATION SCAN

Rank	Frequency (MHz)	Anomaly score	Latent cluster
1	8419.385513	7.298	2
2	8419.385513	6.583	0
3	8419.396759	1.663	1

Band annotation uses an approximate built-in table and should be verified.

VIII. DISCUSSION

Three observations stand out. First, the injection study turns a general expectation into numbers: at a 5% false-alarm rate a classical drift search is far more sensitive than a reconstruction-error score on linear-drift signals, while the CAE needs no drift model at all. Second, on the Kaggle cadences the unsupervised scores keep pace with hand-designed ON–OFF and de-Doppler statistics, which is notable for models that never saw a needle. Third, the taxonomy gives the scores a physical reading: the largest errors concentrate on persistent, interference-like structure, which is consistent with the CAE acting as a sensitive detector of strong recurring features and suggests a natural role as an interference-triage front end ahead of the ON–OFF filter.

The results also point to concrete design implications for learned technosignature detectors.
Score along drift paths. A global maximum of channel-wise error lets noise-driven error compete with faint lines. Integrating error along candidate drift paths, as de-Doppler searches do, would combine the CAE's learned background model with the classical detector's matched filtering.

Make the model cadence-aware. The cadence CAE reconstructs ON and OFF snippets jointly. Supplying the ON–OFF difference as an input channel, or training with an ON–OFF consistency term, would reward exactly the contrast that distinguishes candidates from interference; AE-ON−OFF already ranks highest among the learned scores.
Scale training. The cadence model used 1,200 of roughly 54,000 available needle-free cadences and eight epochs, and its loss was still falling. Training on the full set to convergence, or with a variational objective [6], is the most direct next experiment.

Report at matched false-alarm rates. Comparing learned and classical detectors at the same false-alarm rate, with confidence intervals, makes sensitivity claims checkable. The scripts released with this work make that protocol a one-command run.

IX. SCOPE AND LIMITATIONS

The conclusions apply to the configuration tested. The injection study has 14 trials per SNR level, giving wide intervals; the Kaggle labels mark simulated signals, so the results do not speak to genuine technosignatures; the cadence experiment is label-assisted; and one architecture and one scoring rule were evaluated per setting, on 3,000 of the 60,000 available cadences with a single nominal threshold. Noise and RFI also produce large reconstruction errors, and any real candidate would need verification against interference [11].

X. FUTURE WORK

The framework extends naturally in several directions:

- larger evaluations on additional Breakthrough Listen observations and more of the Kaggle training set;
- deeper convolutional and variational autoencoders and other architectures;
- adaptive thresholding to improve detection while controlling false positives;
- extended baseline comparisons with turboSETI and other detectors on compatible data;
- clustering of similar anomalies to reveal recurring signal patterns;
- setigen injections with different strengths, drift rates and morphologies;
- application of the scanning module to further real radio-astronomy files, checking whether flagged anomalies persist; and
- an interactive dashboard for inspecting spectrograms, scores and evaluation results.

XI. CONCLUSION

This paper benchmarked a convolutional autoencoder for technosignature candidate detection against turboSETI and against non-learned baselines under matched false-alarm conditions. turboSETI recovered every injected signal from SNR 6, while the signal-model-free CAE reached 50% recall at SNR 15. On 1,200 held-out Kaggle cadences, the learned and hand-designed scores all stayed close to chance in the tested configuration (ROC-AUC 0.527–0.556), and the strongest reconstruction errors fell on interference-like structure. The framework supplies quantitative reference points, a diagnostic taxonomy and a reproducible protocol for learned detectors. Drift-aware scoring, ON–OFF-aware inputs and full-scale training are the most promising next steps, and injecting non-linear signal morphologies will test whether learned detectors can find what drift searches cannot.

ACKNOWLEDGMENT

The authors thank the Department of Computer Science and Engineering (Emerging Technologies), SRM Institute of Science and Technology, Vadapalani Campus, for support.

REFERENCES

[1] G. Cocconi and P. Morrison, “Searching for interstellar communications,” *Nature*, vol. 184, no. 4690, pp. 844–846, 1959.
[2] M. Lebofsky et al., “The Breakthrough Listen search for intelligent life: Public data, formats, reduction, and archiving,” *Publ. Astron. Soc. Pac.*, vol. 131, no. 1006, p. 124505, 2019.
[3] Berkeley SETI Research Center, “turboSETI: Narrowband signal detection software.”
[4] G. R. Harp et al., “Machine vision and deep learning for classification of radio SETI signals,” *arXiv preprint* arXiv:1902.02426, 2019.
[5] S. Pardo, D. Poznanski, S. Croft, A. P. Siemion, and M. Lebofsky, “Using anomaly detection to search for technosignatures in Breakthrough Listen observations,” *Astron. J.*, vol. 170, no. 1, p. 12, 2025.
[6] P. X. Ma et al., “A deep-learning search for technosignatures from 820 nearby stars,” *Nature Astron.*, vol. 7, no. 4, pp. 492–502, 2023.
[7] K. Brand, T. L. Grobler, and W. Kleynhans, “CARA: Convolutional autoencoders for the detection of radio anomalies,” *RAS Techn. Instrum.*, vol. 4, p. rzaf005, 2025.
[8] C. Painter et al., “A novel technosignature search in the Breakthrough Listen Green Bank Telescope archive,” *Astron. J.*, vol. 169, no. 4, p. 222, 2025.
[9] J. Li, B. Liang, S. Feng, W. Dai, and S. Wei, “RFI detection based on semi-supervised learning with improved U-Net,” *Astron. Comput.*, vol. 54, p. 101020, 2026.
[10] B. Brzycki et al., “Setigen: Simulating radio technosignatures for the search for extraterrestrial intelligence,” *Astron. J.*, vol. 163, no. 5, p. 222, 2022.
[11] S. Z. Sheikh et al., “Analysis of the Breakthrough Listen signal of interest BLc1 with a technosignature verification framework,” *Nature Astron.*, vol. 5, no. 11, pp. 1153–1162, 2021.
[12] Kaggle, “SETI Breakthrough Listen – E.T. Signal Search.” [Online competition dataset].
[13] Breakthrough Listen, “blimpy: Breakthrough Listen I/O methods for Python.” [Software].
[14] E. B. Wilson, “Probable inference, the law of succession, and statistical inference,” *J. Amer. Statist. Assoc.*, vol. 22, no. 158, pp. 209–212, 1927.
[15] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2015.
[16] J. Davis and M. Goadrich, “The relationship between precision-recall and ROC curves,” in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, 2006, pp. 233–240.
[17] J. A. Hanley and B. J. McNeil, “The meaning and use of the area under a receiver operating characteristic (ROC) curve,” *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

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