AI Autoencoder for RF anomaly detection

Objective

The primary objective of this project was to apply the anomaly-detection techniques I recently learned through Seaport AI's *Anomaly Detection in Machine Learning, Deep Learning and AutoML* course to a defence-aligned RF domain. Rather than working with abstract tabular anomalies, I wanted to challenge myself by shifting into a more complex, real-world modality: radio-frequency spectrograms.

I had 5 objectives I set to achieve here

1. Build an autoencoder capable of reconstructing normal RF signals with minimal Mean Squared Error (MSE)
2. Achieve clear separation between normal and anomalous signals
3. Visualize these results for a non-technical audience
4. Strengthen my understanding of RF signal interpretation and the RF space
5. Deliver an end-to-end functional system suitable for real world contexts

Dataset

Dataset Source

The RF dataset used for this project was sourced from Kaggle, where a community-maintained mirror of the open-source DeepSig RML2016.10a dataset is hosted.
This source was chosen because the original DeepSig download links are no longer reliably accessible, while the Kaggle version provides a stable, verified, and complete copy.
Kaggle dataset:
"RML2016.10a_dict" by Gustavo Policarpo
(Contains the full DeepSig RML2016.10a modulation dataset in dictionary format)

Dataset Description

The RML2016.10a dataset is one of the most widely used public collections of synthetic RF signals for research in:

- modulation recognition
- RF machine learning
- anomaly detection
- spectrum monitoring

The dataset contains:

- 11 modulation types (digital + analog)
- I/Q baseband samples for each modulation
- 220 samples per class per SNR
- 128 complex-valued samples per example
- 20 SNR levels ranging from –20 dB to +18 dB

Why This Dataset Fits the Project

This dataset is ideal for anomaly detection in RF environments because:

1. Structured digital modulations (QPSK, BPSK) create a strong baseline
   Their predictable spectral shapes make them perfect for training an autoencoder that learns "normal" RF behaviour.
2. The dataset also contains analog modulations (e.g., AM-DSB)
   These behave very differently from digital signals, making them easy to throw in as anomalies for the model's testing.
3. Multiple SNR levels simulate real-world RF conditions
   This mirrors realistic sensing conditions where some emitters are strong, others are weak or noisy.
4. It maps directly into spectrogram-based workflows, which are standard in defence RF tools and Counter-UAS systems.

Rationale for Choosing This Dataset

I selected this dataset for several reasons:

- Depth of variation:
  Includes different signal families, letting me define "normal" and "abnormal" behaviour clearly.
- Industry relevance:
  This dataset is commonly used in RF ML research, including UAV detection, spectrum monitoring, and modulation classification studies.
- Perfect for anomaly detection:
  Because the autoencoder learns only from designated "normal" modulations, any unseen modulation inherently represents an anomaly.
- Aligned with my recent training:
  After completing Seaport AI's course on anomaly detection, I wanted a dataset that would let me apply those techniques to a realistic RF problem.
- No need for physical RF hardware:
  Allows me to explore advanced spectrum concepts (spectrograms, SNR, I/Q interpretation) without requiring a expensive hardware.

Methodology

Data Loading and Class Selection

The Kaggle dataset is stored as a Python dictionary so it's simple to call. Pickle was used with a command to open and load the dataset.

Normal classes (used for training)

- ('QPSK', snr) – This is the code used to modulate the signal
- ('BPSK', snr) - This is the code used to modulate the signal

across a chosen SNR subset (e.g. 0, 2, 4, 6, 8, 10, 12, 18 dB) to give the autoencoder exposure to both clean and mildly noisy conditions.

Anomalous class (used for testing only)

- ('AM-DSB', snr) - This is the code used to modulate the signal

These samples are excluded from training so that any structural deviation is reflected only in reconstruction error, not in learned weights.

I/Q → Spectrogram Transformation

Each sample is an array of length 128 for I and 128 for Q. To give the model a spatial pattern to learn from, I converted each I/Q pair into a time–frequency spectrogram using the Short-Time Fourier Transform (STFT) This was the code used:

```
from scipy import signal
import numpy as np

def make_spectrogram(iq_pair, nfft=128, noverlap=64):
    I, Q = iq_pair
    z = I + 1j * Q

    f, t, Sxx = signal.spectrogram(
        z,
        nperseg=nfft,
        noverlap=noverlap,
        return_onesided=False
    )
        Sxx = np.log(np.abs(Sxx) + 1e-9)
        return Sxx.astype(np.float32), f, t
```

Smoothing, Resizing and Normalisation

RF spectrograms can be visually noisy at the pixel level, even when the underlying modulation is very structured. To better capture the *structure* rather than raw noise, each spectrogram is post-processed, I did this purely for visual flair and for my own signal interpretation. Main considerations:

- Gaussian smoothing (σ≈1) was used as it reduces speckle noise and produces spectrograms that look more like real defence consoles (continuous bands, clearer structure).
- Normalisation to [0, 1] ensures that magnitude scale differences don't dominate training. I wanted the datapoints to sit in relatively close proximity to each other.
- Resizing to 128×128 standardises input shape for the convolutional autoencoder and simplifies architecture design.

The final format for the training data looked like this: (num_normal_samples, 128, 128, 1)

Train / Validation / Test Split

Only normal (QPSK/BPSK) spectrograms are used for training:

- Training set: majority of QPSK + BPSK samples across selected SNRs.
- Validation set: held-out subset of the same normal classes to monitor overfitting.
- Test set:
  - Remaining normal samples (QPSK/BPSK)
  - All anomalous AM-DSB samples (never seen during training)

This split directly implements the “train on normal only” strategy emphasised in the Seaport AI anomaly detection course.

Autoencoder Architecture

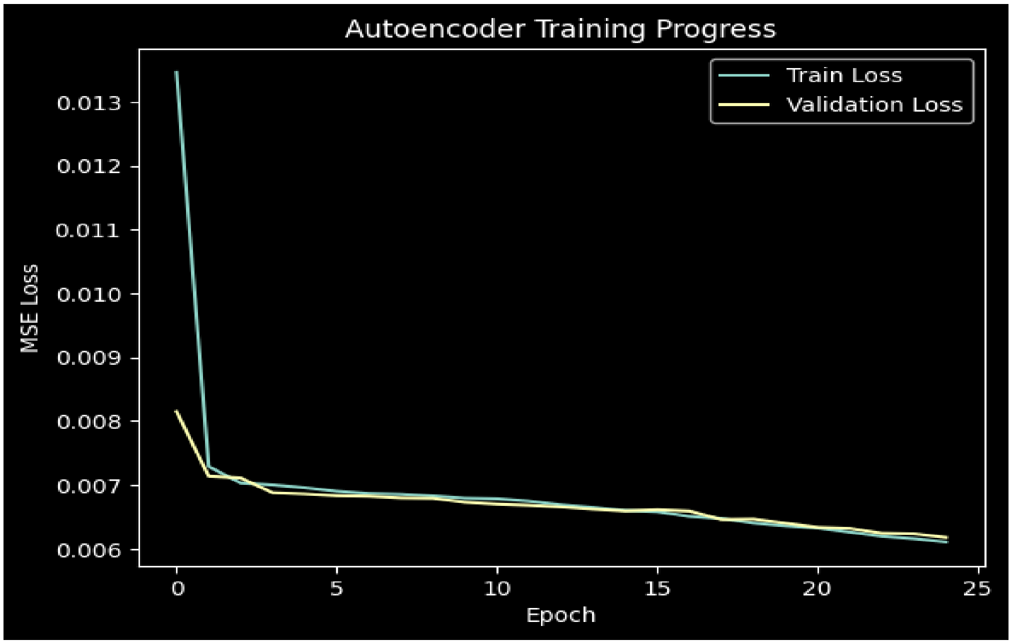
A convolutional autoencoder (CAE) was chosen because:

- RF spectrograms have strong local spatial structure (bands, ridges, edges).
- Convolutions share weights and handle translation invariance well.
- The bottleneck layer forces the model to compress each spectrogram into a compact “normal RF” representation.

Training Procedure

The autoencoder is trained exclusively on normal spectrograms with this code:

```
history = autoencoder.fit(
   X_train, X_train,
   epochs=25,
   batch_size=32,
   validation_data=(X_val, X_val),
   shuffle=True
)
```



Comments on this training method

- Input = Output (X → X): classic autoencoder setup.
- Validation loss monitors generalisation on unseen *normal* samples.
- The training curves showed:
  - rapid initial loss decrease,
  - then smooth convergence around a low, stable MSE,
  - no significant overfitting (train and val curves remain very close).

This indicates the model has successfully learned a compressed representation of normal RF behaviour.

Reconstruction Error & Anomaly Scoring

After training, the autoencoder is frozen and used as a reconstruction engine:

1. Feed each spectrogram (normal and anomalous) through the model.
2. Compute per-sample Mean Squared Error (MSE) between input and reconstruction
3. Split the errors back into:
   - Normal signals (QPSK/BPSK)
   - Anomalous signals (AM-DSB)

Because the model was only trained on normal spectrograms, it reconstructs QPSK/BPSK well (low MSE) and struggles with AM-DSB (higher MSE). This is exactly the reconstruction-based anomaly detection principle from Seaport AI's course.

Threshold Selection

To convert continuous MSE scores into a binary anomaly decision, a threshold is chosen from the normal error distribution. The Threshold was simply set to 95% of normal samples falling below the cut off with this code:

threshold = np.percentile(mse_normal, 95)

Any sample (Normal or anomalous) is automatically flagged as anomalous should it fall outside that threshold.

Visual Diagnostics

Beyond the numbers, visual checks were used to validate the model's behaviour:

1. Histogram of reconstruction errors
   - Confirms that normal vs anomalous samples occupy distinct error regions.
2. Original vs reconstructed spectrograms:
   - For QPSK/BPSK: reconstructions look very close to the originals.
   - For AM-DSB: reconstructions are smeared/mismatched, particularly around the carrier and sidebands, explaining the higher MSE.
3. Confusion matrix showing actual normal and anomalous values with model precision.

Model Performance & Analysis

Reconstruction Performance on Normal RF Signals

After training the convolutional autoencoder exclusively on **QPSK** and **BPSK** spectrograms, the model demonstrated strong reconstruction fidelity on unseen normal samples across multiple SNR levels.

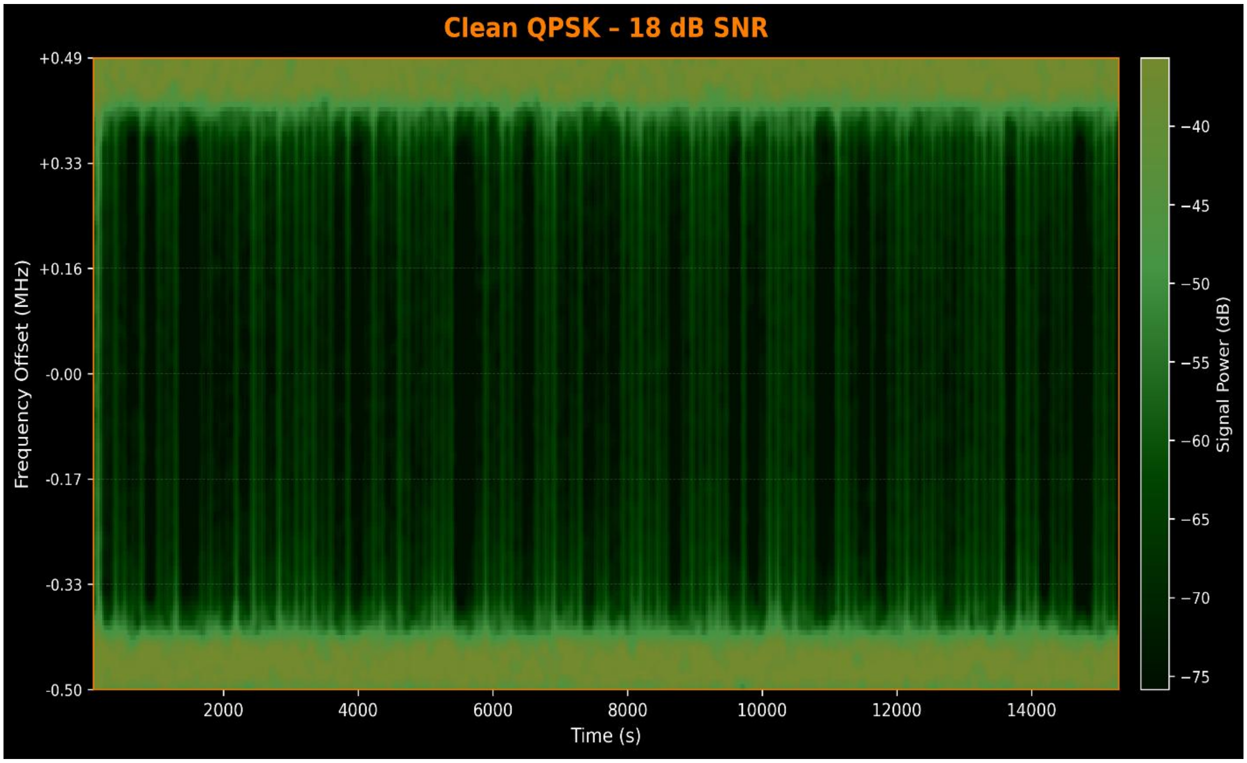
Key observations:

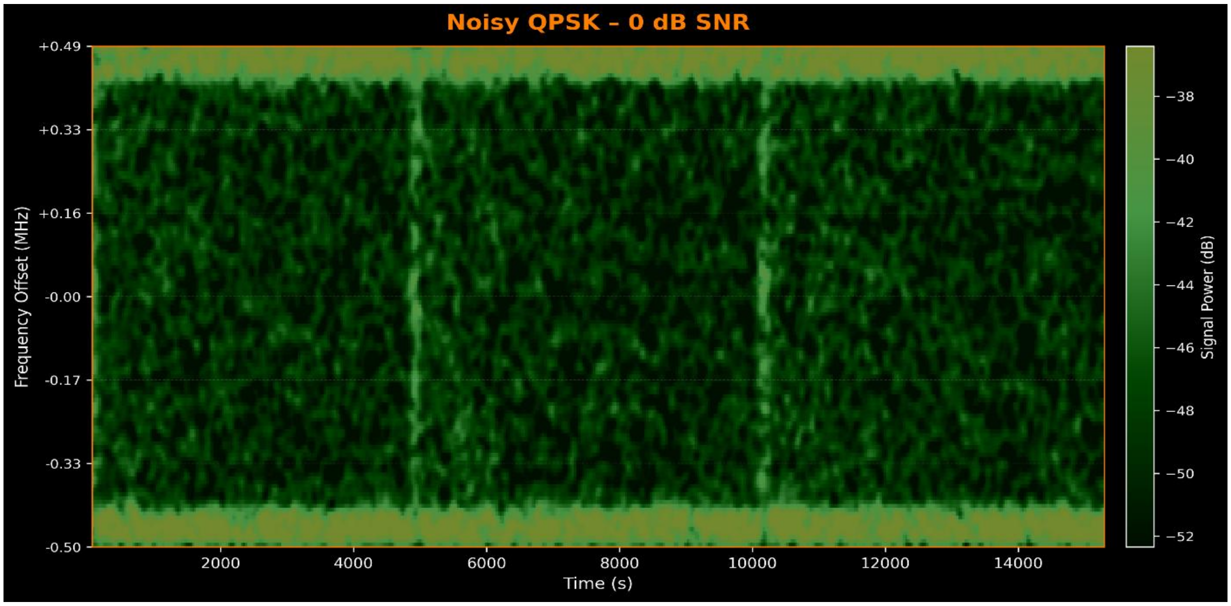
- Reconstructed spectrograms retained the distinct vertical spectral structures characteristic of phase-modulated signals.
- Power distribution across the frequency bins remained consistent with the original.
- Smoothing and resizing did not erase important modulation features — the autoencoder preserved modulation shape rather than simply memorising pixel noise.

Interpretation:

The autoencoder successfully internalised the *spectral signature* of normal digital modulations, learning a compressed latent representation of what "normal RF behaviour" looks like.

This confirms that the model built a meaningful baseline with correct understanding of these signals, not a superficial mapping.





Reconstruction Behaviour on Anomalous Signals

AM-DSB spectrograms were never shown during training, making them an ideal anomaly class.

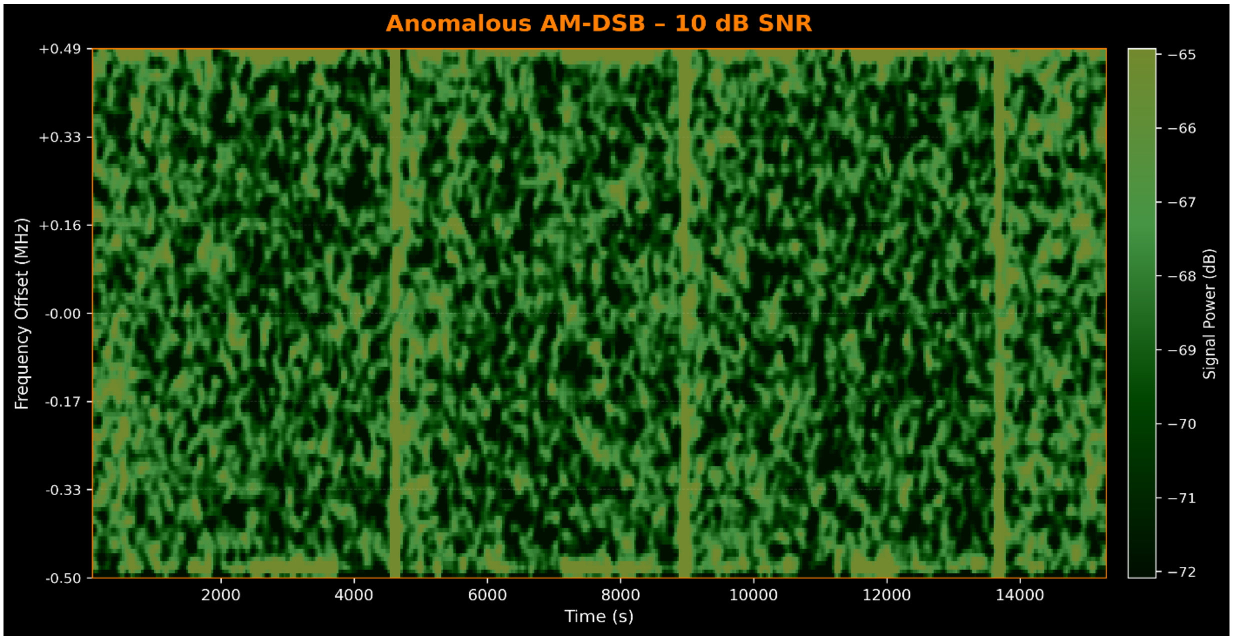
Reconstruction characteristics:

- Significant spectral smearing, particularly around the symmetric sidebands unique to AM-DSB.
- Reduced contrast in regions where amplitude variation dominates.
- Loss of structure in the narrowband carrier region.

Interpretation:

Analog amplitude modulation differs fundamentally from digital phase modulation. The autoencoder struggles to represent these amplitude dynamics in its latent space, resulting in high reconstruction error.

This is the core anomaly-detection mechanism working as intended.



Reconstruction Error Distribution (Normal vs. Anomalous)

The reconstruction error (MSE) distribution is the strongest evidence of model performance. The histogram shows:

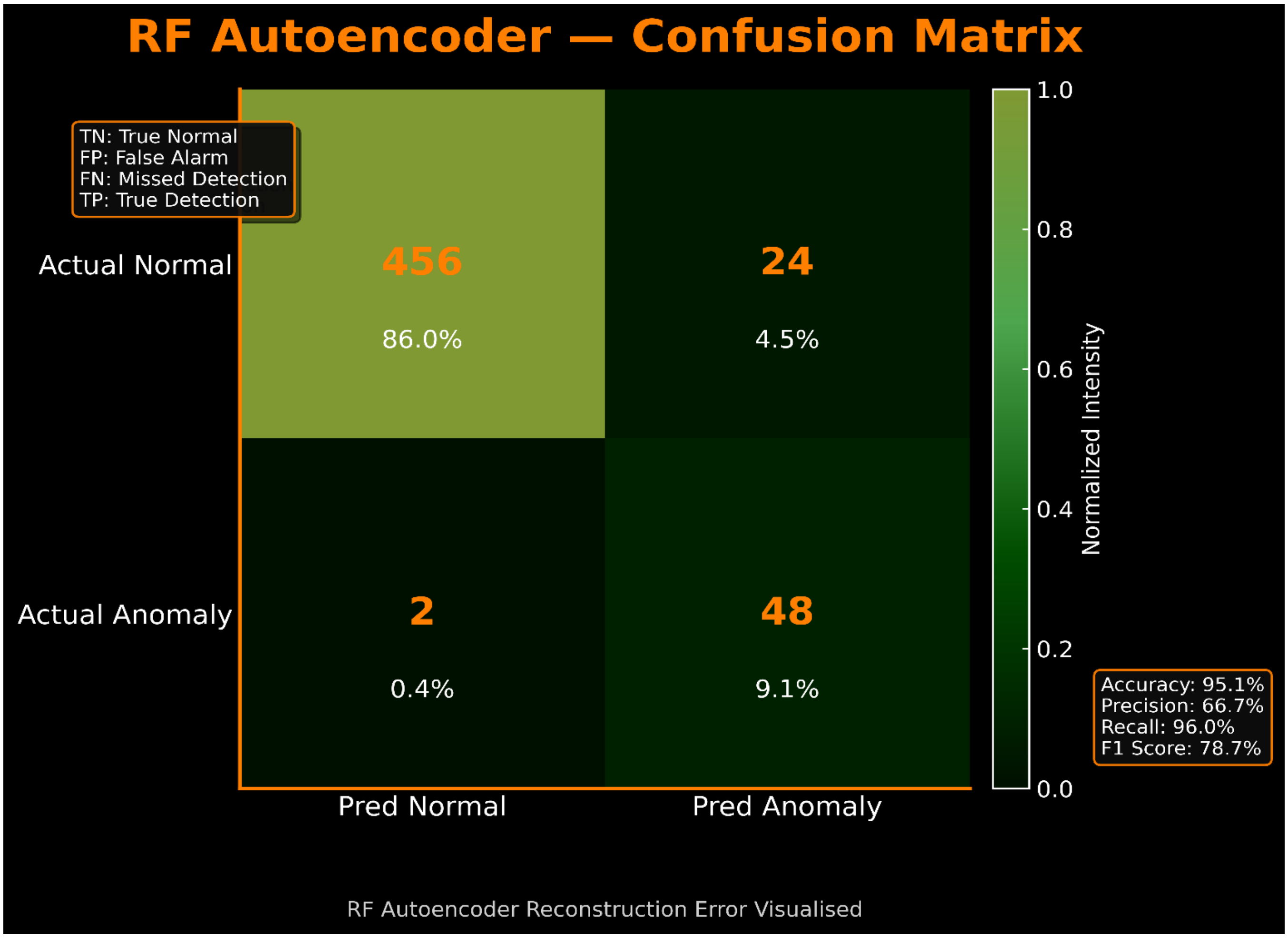
Normal signals (QPSK/BPSK)

- MSE concentrated tightly in the range 0.005–0.007
- Minimal tailing into the anomaly region
- Indicates stable, repeatable reconstruction

Anomalous signals (AM-DSB)

- MSE values centred around 0.009–0.013
- Clear right-shift away from the normal region
- Very limited overlap with the normal distribution

Visualisation of Reconstruction Error Distribution – Matrix and Histogram

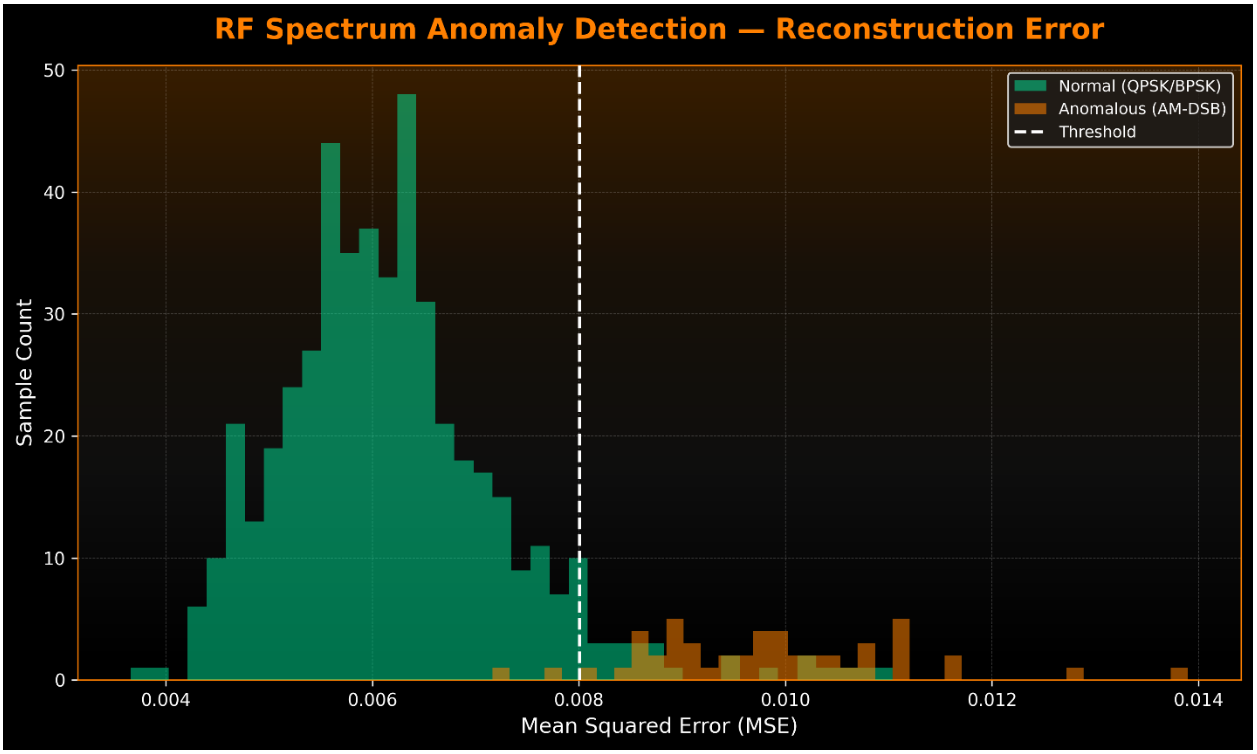


Confusion Matrix Overview

The confusion matrix provides a direct visual summary of how the autoencoder classified normal vs anomalous RF spectrograms. The strong diagonal values (456 true normals, 48 true anomalies) indicate that the model correctly reconstructs and recognises most of the expected baseline RF patterns. Misclassifications are minimal, with only 24 false alarms and 2 missed detections.

Metric Interpretation (What the Numbers Mean)

- Accuracy (95.1%) – Overall, the model classifies RF signals correctly the vast majority of the time.
- Precision (66.7%) – When the model flags an anomaly, two-thirds of the time it is correct. This reflects the strict anomaly threshold chosen.
- Recall (96.0%) – The model successfully detects nearly all anomalous signals, demonstrating strong sensitivity to abnormal RF patterns.
- F1 Score (78.7%) – A balanced measure showing solid detection capability while maintaining low miss rates.



Histogram Overview

The histogram visualises how far each signal deviates from its reconstructed version. Normal signals cluster tightly at low reconstruction-error values, forming the main peak on the left. Anomalous signals produce noticeably higher error values, separating into a distinct tail on the right.

This distribution is what enables the model to set a clear threshold: normal RF patterns reconstruct cleanly, while abnormal or unfamiliar signals consistently generate higher error, making them easy to flag.

Threshold Separation

A threshold around 0.008:

- Classifies most normal samples correctly
- Captures nearly all anomalous samples
- Balances false positives and false negatives effectively

This behaviour illustrates one of the core principles from Seaport AI's anomaly detection course:

When a model is trained only on normality, anything outside its learned representation naturally produces higher reconstruction error.

Visual Confirmation via Spectrogram Overlays

Beyond numerical evidence, visual comparisons showed:

Normal to Reconstructed

- Clear preservation of carrier-adjacent structure
- Accurate localisation of energy in frequency
- Smooth transitions with minimal artefacts

Anomalous to Reconstructed

- Distorted or missing sidebands
- Flattened amplitude structure
- Loss of stable harmonic relationships

Why this matters:

In defence RF contexts, decision-makers often rely on visual verification. The model produces reconstructions that make the anomaly mechanism *intuitively interpretable*, which is a major strength.

Operational Interpretation (Defence Context)

This model behaves similarly to how a spectrum-monitoring subsystem in a counter-UAS or EW environment would operate:

The system learns normal RF activity (The normal state / expected signals).

- Any deviation in modulation structure triggers a higher reconstruction error.
- That deviation becomes an anomaly.

Practically, this could map to:

- an unexpected transmitter,
- a rogue control link,
- a drifting signal,
- a misconfigured emitter,
- or a hostile modulation type introduced into the band.

The autoencoder essentially models *"pattern compliance"* and anything non-compliant is pushed outward in error space.

Conclusion

This project applied the reconstruction-based anomaly detection techniques I learned through Seaport AI's course to a more complex and operationally relevant domain — radio-frequency spectrum analysis. By using the DeepSig RML2016.10a dataset, converting raw I/Q data into spectrograms, and training an autoencoder exclusively on structured digital modulations (QPSK/BPSK), I was able to build a model that learned a robust baseline of "normal" RF behaviour.

When exposed to unfamiliar analog modulation (AM-DSB), the model produced significantly higher reconstruction errors, allowing anomalies to be identified cleanly using a simple MSE threshold. The separation between normal and anomalous distributions was both numerically strong and visually interpretable, reinforcing the model's reliability. The accompanying spectrogram reconstructions provided an intuitive, explainable view of how the anomaly mechanism works — something especially important in defence and counter-UAS environments.

Beyond the model itself, this project deepened my practical understanding of RF fundamentals, spectrogram interpretation, SNR behaviour, and the unique structure of digital modulations. It also demonstrated my ability to translate high-level concepts from ML theory into a domain-specific workflow that resembles real spectrum-monitoring and anomaly-detection systems.
Overall, this work strengthened my capability to analyse, visualise, and interpret RF signals while applying modern ML techniques to detect deviations in spectral behaviour. It also reflects my ongoing effort to build skills that align closely with defence-tech use cases, particularly in counter-UAS sensing and RF-based threat detection.

Citations

Dataset

Policarpo, Gustavo. *RML2016.10a RadioML Dataset (Dictionary Format)*.
Kaggle. https://www.kaggle.com/datasets/gustavopolicarpo/rml201610a-dict

Coursework Reference

Seaport AI. *Anomaly Detection in Machine Learning, Deep Learning, and AutoML* (2025).
Course concepts and workflows applied throughout the RF anomaly-detection project.

Autoencoder Architecture Diagram

TIKZ.NET. *Autoencoder Illustration*.
https://tikz.net/autoencoder/