V1

Chunyu Yang

February 15, 2025

1 Introduction

The rapid proliferation of non-geostationary orbit (NGSO) satellite constellations, driven by advancements in low-cost launch technologies and the demand for global broadband connectivity, has intensified the risk of co-frequency interference with geostationary orbit (GSO) systems. For instance, large-scale NGSO constellations like Starlink and OneWeb share overlapping frequency bands with critical GSO services such as fixed-satellite (FSS) and broadcasting-satellite (BSS) systems, creating unprecedented challenges in spectrum coexistence. The International Telecommunication Union (ITU) underscores the urgency of interference mitigation in Radio Regulation, which mandates that NGSO systems ”shall not cause unacceptable interference to geostationary-satellite networks” while imposing stringent limits on equivalent power flux density (EPFD) and off-axis emissions. These regulations aim to protect GSO operations, such as weather monitoring and telecommunications, which underpin vital global infrastructure.

2 System Model

This study explores the downlink dynamics of forward links originating from two distinct satellite constellations. We focus primarily on satellites in geostationary orbit (GSO), serving as the main communication system. Satellites in non-geostationary orbit (NGSO), particularly those in low Earth orbit (LEO), are considered potential sources of interference to a general GSO ground station (GGS). In this section, we provide a comprehensive description of the system model, which encompasses link budget calculations and the modeling of the received digital baseband signal, shown in Figure 1

2.1 Link Budget Model

In our analysis, both GSO and LEO satellites are actively transmitting signals, with their main transmission lobes consistently directed toward a GGS. At the GGS receiver, the carrier power received from the desired GSO satellite system can be calculated using the following equation:

$$C = \frac{\mathrm{EIRP}_{\mathrm{gso}} \times G_{\mathrm{r,\ gso}(\theta_0)}}{L_{\mathrm{FS,\ gso}} \times L_{\mathrm{add}}}$$

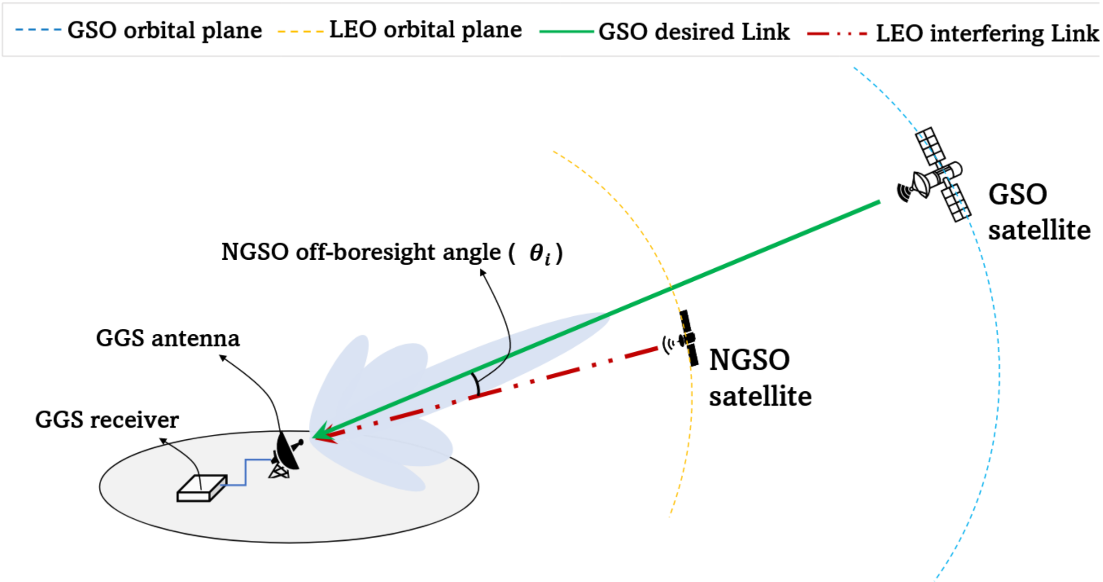

Figure 1: Interference model where the primary system is GSO satellites and the interference source is LEO satellites.

where C is the received carrier power, EIRP_{gso} is the equivalent isotropic radiated power of the GSO satellite, $G_{\text{r, gso}(\theta_0)}$ is the maximum receive antenna gain of the GGS, with θ_0 being the boresight angle. $L_{\text{FS, gso}}$ is the free space path loss (FSPL) between the GSO satellite and the GGS. L_{add} accounts for additional losses in the link, such as atmospheric and environmental losses.

As LEO satellites traverse the visible sky from the GGS, the station may concurrently receive signals from one or more LEO satellites, especially when they operate in the same frequency bands as the GSO satellite. These LEO signals can elevate the interference level within the primary signal. The interference power I_k received from an individual LEO satellite indexed by k is calculated as:

$$I_k = \frac{\text{EIRP}_k \times G_{\text{r, k}}(\theta_k) \times B_{\text{adj, k}}}{L_{\text{FS, k}} \times L_{\text{add}}}$$

where $B_{\text{adj, k}}$ is the adjustment factor accounting for bandwidth overlap between the GSO and LEO signals, with a value between 0 and 1.

The downlink carrier-to-noise ratio (CNR) of the GSO satellite signal received by the GGS is calculated as:

$$\text{CNR} = \frac{C}{k_{\text{boltz}} T B}$$

where k_{boltz} is the Boltzmann constant, T is the receiver noise temperature., and B is the bandwidth of the GSO baseband signal.

Similarly, the interference-to-noise ratio (INR) received from the kth LEO satellite is:

$$\text{INR}_k = \frac{I_k}{k_{\text{boltz}} T B}$$

To evaluate the effect of interference from all LEO satellites, we use the carrier-to-interference-plus-noise ratio (CINR):

$$\text{CINR } = \frac{C}{\sum_{k=1}^{K} I_k + k_{\text{boltz}} T B}$$

where K is the total number of interfering LEO satellites at a given time.

2.2 Received Signal Model

The total received signal at the GGS is the sum of the desired signal, interference from all LEO satellites, and additive white Gaussian noise (AWGN):

$$y(t) = x(t)\sqrt{\text{CNR }} + \sum_{k=1}^{K} \left(i_k(t) e^{j2\pi(f_{\text{c, k}} - f_{\text{c, gso}})t} \sqrt{\text{INR }_k} \right) + \zeta(t)$$

where $x(t)$ is the baseband signal trasmitted by GSO satellite, $i_k(t)$ is the interfering signals from the kth LEO satellite. At give time t, there are K interfering LEO satellites, the carrier frequencies of the kth LEO satellite and the GSO satellite are $f_{\text{c, k}}$ and $f_{\text{c, gso}}$ respectively. The exponential term accounts for the frequency difference .

For further analysis and potential machine learning applications, the received signal can be represented in both time and frequency domains:

1. Time-domain representation: The amplitude of $y(t)$ is sampled to create a one dimensional sequence y^A.

2. Frequency-Domain Representation: The power spectral density (PSD) of $y(t)$ is estimated using Welch's method, which divides the signal into overlapping segments and computes the periodogram of each segment, resulting in one dimensional $y^F = 10\log_{10}(\phi(y(t)))$

3 Proposed Deep Learning Models

This section presents a novel framework for multi-modal interference detection in satellite communication, leveraging both time domain and frequency domain data. Three classes of deep learning models are proposed: Autoencoders(AE), Variational Autoencoders(VAE), and Conditional Variational Autoencoders(CVAE). Each class is implemented with three architectural variants: linear, convolutional and transformer based, to explore the impact of feature extraction mechanisms. The core innovation lies in fusing time and frequency information to anomaly detection robustness.

Next, we seek to define a reliable reconstruction error threshold. This threshold will then be applied to signals, regardless of interference. If the detected error values surpass this set threshold, interference is identified.

We denote the time and frequency domain signal input as S_t and S_f. Before being fed into the model, each input undergoes elementwise normalization, scaling values between 0 and 1 using the minimum and maximum values from the relevant training dataset. The reconstructed output is denoted as $\hat{S}_t$ and $\hat{S}_f$.

1. Linear: TODO: How Linear works

2. CNN: The CNN based class takes in a one-dimensional signal and processes it through a series of convolutional layers. The first convolutional layer extracts local temporal patterns, expanding the feature space from a single channel to 16 channels. A ReLU activation introduces non-linearity, helping the model capture complex relationships in the data. After this, a max pooling layer reduces the dimensionality, improving computational efficiency while retaining essential features. The second group increases channel number to 32. The output is then flattened to a two dimensional tensor of shape [B, E]. The frequency-domain encoder follows an identical architecture. The decoder reverses the encoding process to reconstruct the original signals from the fused latent representation. Two transposed convolution layers progressively upsample the features, restoring the signal's original resolution. ReLU activations ensure non-linearity, aiding reconstruction. The final output is the recovered time-domain signal and frequency-domain spectrum.

3. Transformer: TODO: How transformer works

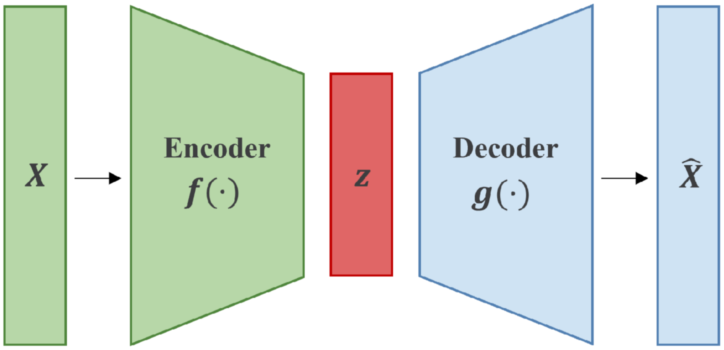

Figure 2: Autoencoder(VAE) class

The AE class is a fundamental deep learning model that learns a compressed representation of the input data. The AE consists of an encoder and a decoder, which are trained to minimize the reconstruction error between the input and output. First a dual branch is used to map S_t and S_f to a latent dimension E. Then these latent representations are concatenated and have a dimension of $2E$. An MLP layer is then applied to map the fused vector into dimension E, which is the shared bottleneck of both time and frequency domain data's latent space. Finally, two domain specific decoders are used to reconstruct the time and frequency domain signals. In the case of linear autoencoder where we also drop the non-linear operations, the autoencoder would achieve the same latent representation as Principal Component Analysis (PCA). The architecture is shown in Figure 2.

To model the reconstruction error, we use the mean squared error (MSE) loss function:

$$L_{\mathrm{AE}} = \frac{1}{N} \sum_{i=1}^{N} \left(\left\| S_t^{(i)} - \hat{S}_t^{(i)} \right\|_2^2 + \left\| S_f^{(i)} - \hat{S}_f^{(i)} \right\|_2^2 \right)$$

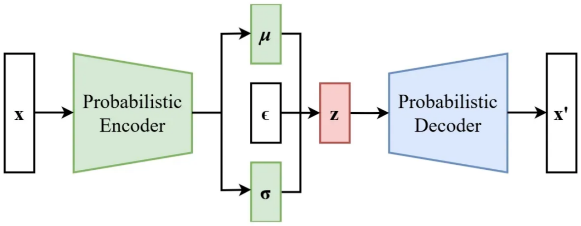

Figure 3: Variational Autoencoder(VAE) class

The Variational Autoencoder (VAE) extends traditional autoencoders by introducing a probabilistic framework to model the latent space. Unlike deterministic autoencoders, VAEs learn a generative model of the data by approximating the true posterior distribution of latent variables z given the input x. This is achieved via an encoder-decoder architecture governed by parameters ϕ(encoder) and θ (decoder), as shown in Figure 3.

The objective is to maximize the evidence lower bound :

$$\text{ELBO}(\theta, \phi; x) = \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x|z)] - D_{\text{KL}} \left(q_\phi(z|x) || p(z)\right)$$

Our VAE implementation extends this framework to fuse time-domain S_t and frequency domain S_f signals. Two domain specific encoders process S_t and S_f, produce h_t and h_f. Features are concatenated and mapped to latent parameters:

$$\mu = f_\mu(h_t, h_f) \quad \sigma = f_\sigma(h_t, h_f)$$

The latent vector z is sampled as $z = \mu + \sigma \cdot \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$ to enable backpropagation. The decoder then reconstructs the input signals from the latent representation. The loss combines reconstruction loss and KL divergence :

$$L_{\text{VAE}} = \text{MSE}(S_t, \hat{S}_t) + \text{MSE}(S_f, \hat{S}_f) - \beta \cdot \frac{1}{2} \sum_{i=1}^{d} (1 + \log \sigma_i^2 - \mu_i^2 - \sigma_i^2)$$

The hyperparameter β inspired by TODO: Inspired by what encourages disentangled representations by amplifying the KL term's regularization effect.

4 Experiment Results

4.1 Data Preprocessing & Dataset Configuration

Our experiments leverage the satellite interference dataset from Saif Al-Dawla et al. [Sai+23], comprising 17,281 time-domain magnitude sequences captured from Geostationary (GSO) and Low-Earth Orbit (LEO) satellite links. Table 1 details the dataset splits adhering to the original paper's rigorous simulation setup:

Each sample consists of an 800-dimensional time-series magnitude vector derived from DVB-S2X standard transmissions at 100 MHz sampling rate. We preprocess the data as follows:

Table 1: Dataset Composition and Splits [Sai+23]

Split	**Total Samples**	**Clean (Class 0)**	**Interference (Class 1)**	**SNR Range (dB)**
Train	11,509	11,509	0	6.4–15.4
Validation	1,302	1,302	0	6.4–15.4
Test	4,470	2,235	2,235	4.5–32.5

Normalization: To mitigate amplitude variations from satellite link adaptation, we apply instance-wise standardization:

$$\hat{x}i = \frac{x_i - \mu}{\sigma}$$

$$\mu = \frac{1}{800} \sum k = 1^{800} x_k$$

$$\sigma = \sqrt{\frac{1}{800} \sum_{k=1}^{800} (x_k - \mu)^2}$$

4.2 Models evaluation results

The proposed framework is rigorously evaluated against a conventional Energy Detection (ED) baseline, which computes the energy threshold β_E by optimizing the AUC score on the test dataset. For a received signal $\mathbf{y}_n$, the energy $E_n = \sum_{i=1}^{N} |y_n[i]|^2$ is compared to β_E to detect interference. The ED method achieves 71.67% accuracy and 0.7167 AUC, with perfect precision (100.0%) but critically low recall (43.35%), reflecting its propensity for false negatives under non-Gaussian noise conditions.

Our proposed models—LinearAE, CNNAE, TransformerAE, LinearVAE, CNNVAE, and TransformerVAE—leverage fused time-frequency domain inputs to address these limitations. Architectural variants span linear, convolutional, and transformer-based feature extractors, enabling a systematic exploration of anomaly detection robustness.

As shown in Table 2, CNNAE achieves the highest AUC (0.9175) among all models, surpassing the ED baseline by 20.08% and demonstrating the efficacy of convolutional architectures in capturing localized interference signatures. The deterministic autoencoder (AE) class consistently outperforms variational autoencoders (VAE), with CNNAE exceeding CNNVAE by 3.51% in AUC, suggesting that probabilistic latent spaces may introduce unnecessary complexity for this task.

The LinearAE model, despite its simplicity (4.20M parameters), delivers competitive performance (AUC: 0.9176), highlighting the sufficiency of linear projections for interference detection in fused representations. However, its marginally lower recall (81.81% vs. CNNAE: 81.68%) underscores the value of hierarchical feature learning in mission-critical scenarios.

In the context of architectural efficiency, LinearAE (4.20M parameters) and LinearVAE(2.4M parameters) achieve near-instant inference (0.0216–0.0253 thresholds), ideal for edge devices with stringent latency constraints. CNNAE (2.47M parameters) and CNNVAE

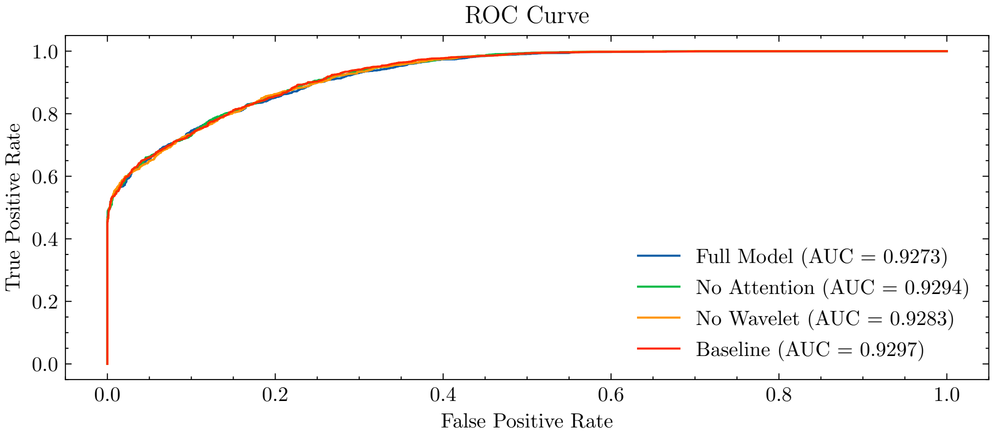

Figure 4: Receiver Operating Characteristic (ROC) curves for AE and VAE models

(2.50M) balance performance and efficiency, requiring 60% fewer parameters than transformer variants while delivering superior AUC.

As for training dynamics: Transformer architectures exhibit pronounced instability: TransformerAE collapses entirely (AUC: 0.6690), with training loss oscillations indicating poor convergence. TransformerVAE struggles to optimize its 27.50M parameters, achieving subpar recall (78.64%) despite extensive training.

Transformer-based models, despite their theoretical allure, fall short in this context. The TransformerVAE, with its hefty 27M parameters, only manages an 87.3% AUC, highlighting a clear mismatch between its complexity and the task's requirements. Even more concerning is the TransformerAE's convergence issue, which drags its AUC down to a mere 66.9%. These results suggest that the lack of a strong inductive bias for sequential data makes transformers less suited for interference detection in this setting, where localized features play a dominant role.

Table 2: Classification Performance of AE and VAE Models

Model	**Threshold**	**Precision**	**Recall**	**F1-Score**	**Accuracy**	**AUC**	**Params(M)**
Autoencoders (AE)							
LinearAE	0.0216	0.8182	0.8181	0.8181	0.8181	0.9176	4.20
CNNAE	0.0208	0.8168	0.8168	0.8168	0.8168	0.9175	2.47
TransformerAE	0.0013	0.5278	0.5277	0.5277	0.5277	0.6690	10.96
Variational Autoencoders (VAE)							
LinearVAE	0.0253	0.7895	0.7895	0.7895	0.7895	0.8860	2.40
CNNVAE	0.0255	0.7891	0.7890	0.7890	0.7890	0.8864	2.50
TransformerVAE	0.0325	0.7864	0.7864	0.7864	0.7864	0.8733	27.50

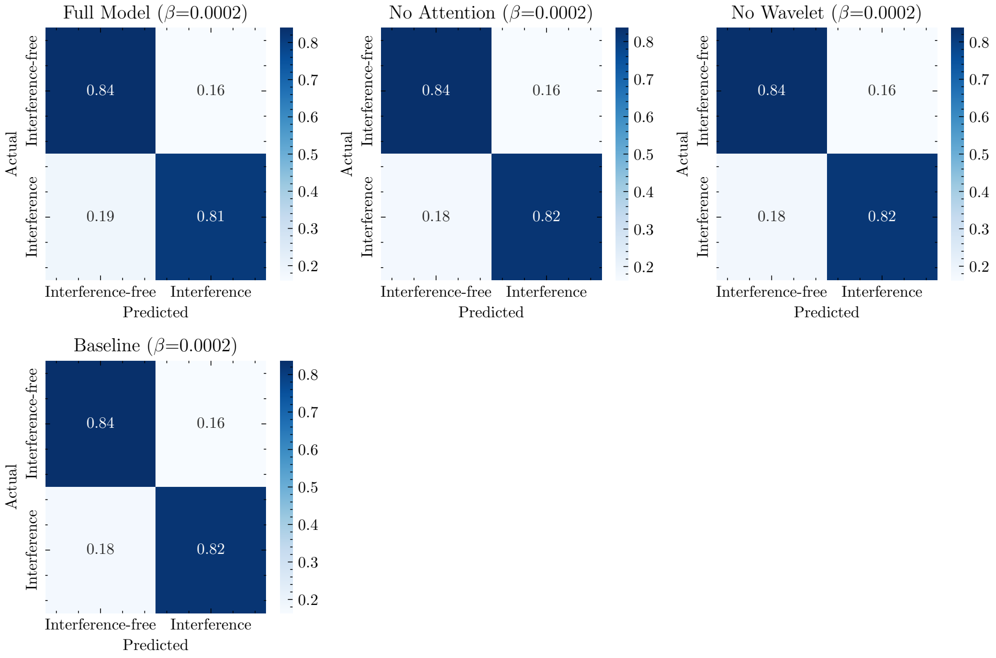

Figure 5: Confusion matrix for AE and VAE models

References

[Sai+23] Almoatssimbillah Saifaldawla et al. “NGSO-To-GSO Satellite Interference Detection Based on Autoencoder”. In: *2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. 2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC). Sept. 2023, pp. 1–7. DOI: `10.1109/PIMRC56721.2023.10293839`. URL: `https://ieeexplore.ieee.org/document/10293839/?arnumber=10293839` (visited on 10/04/2024).