Convolutional Autoencoders for Non-Geostationary Satellite Interference Detection

Almoatssimbillah Saifaldawla, Flor Ortiz, Eva Lagunas and Symeon Chatzinotas
Interdisciplinary Centre for Security Reliability and Trust (SnT),
University of Luxembourg, Luxembourg
e-mail: {moatssim.saifaldawla, flor.ortiz, eva.lagunas, symeon.chatzinotas}@uni.lu

***Abstract*—The new wave of mega-constellations in low-earth orbits (LEOs), which occupy frequency bands that are also used by legacy geostationary orbit satellites (GSOs), will eventually result in increased interference events. In this paper, we explore the application of unsupervised deep learning models, specifically convolutional autoencoders (CAEs), for detecting non-Geostationary Orbit Satellite (NGSO) interference in GSO ground stations (GGS). We propose and evaluate two DL models, the CAE1D model handling amplitude values as 1D data, and the CAE2D model handling In-phase/Quadrature (IQ) samples as 2D data. Through rigorous experimentation, we examine the models' performance against traditional energy detector (ED) methods, employing single-model (SM) and multi-models (MMs) approaches for training the models. Our findings reveal that DL models, particularly under the MMs approach, significantly outperform conventional methods with up to 11% in the probability of detecting interference, demonstrating the potential of advanced machine learning techniques to improve the reliability of satellite communication systems, especially for such fast-varying interference environments (LEO mobility), where only GSO interference-free signals are needed for training the models.**

***Index Terms*—GSOs, NGSOs, Interference Detection, Satellite Communication, Convolutional Autoencoders (CAEs).**

I. Introduction

The recent development of large constellations of non-geostationary orbit (NGSO) satellites, especially in low-Earth orbits (LEO), addresses the significant challenge of connecting almost 37% of the global population who currently lack Internet access [1]. The proliferation of LEO orbit satellites raises substantial risks of unintentional interference with legacy geostationary orbit satellites (GSOs) communications since both systems have been assigned to the same frequency bands [2]. In this paper, we investigate the interference detection problem as it lays the basis for interference management.

The need to enhance both the intricacy and power utilization aspects of conventional interference detection processes has led to an increased interest in the potential application of advanced machine learning (ML) and deep learning (DL) strategies as alternatives [3]. One ML/DL model can learn from interference-free data , and then automatically detect various types of interference signals, thus reducing the requirement for separate units for each interference type without needing any prior knowledge to detect interference. Pellaco et al. [3] employed a Long Short-Term Memory (LSTM) model to identify interferences in the NGSO satellite signal spectrum observed at GSO ground stations (GGSs). This method was specifically designed to pinpoint intentional jamming incorporated into NGSO communication. Vazquez et al. [4] introduce a convolutional autoencoder (CAE) model to detect interference signals in satellite communication, by processing in-phase and quadrature (IQ) samples. Their study focused on high-level exploration of ML for interference detection. Saifaldawla et al. [5] developed a DL model based on autoencoder (AE) for automatic detection of NGSO system interference with GSO-GGS communication. Their study concentrated on a singular modulation scheme (ModCod) applied to both GSO and NGSO systems.

In this paper, we utilize advanced DL models capable of detecting NGSO interference signals in GSO signals at GGS, which were simulated using a realistic satellite scenario explained within the paper. Our models incorporates the convolutional process into the AE-based design, enhancing its ability to extract features from input data that have spatial or temporal patterns. We introduced DL models that are capable of detecting interference in 1D or 2D data, with less complex architectures and more detection capabilities compared to the one in [4]. This paper main contributions can be outlined as follows,

- We design, develop, and implement two distinct DL models, 1) one model is dedicated for 1D data, or the magnitude representation of the signal, namely, CAE1D. 2) The second model is dedicated for 2D data, or the I/Q representation of the signal, namely, CAE2D. By comparing CAE1D versus CAE2D we can observe which representation is better at detecting interference.
- We investigate which approach of model training is more promising to detect interference, 1) Having one model trained with all GSO ModCods data combined, namely single-model (SM) approach. 2) Having many models, each trained with specific GSO ModCod, namely, multi-models (MMs) approach. By evaluating both approaches, we can observe which one is given better interference detection accuracy.
- We conducted a thorough comparative interference detection analysis between our newly developed DL models and traditional energy detector (ED) method. This comparison highlights the strengths and potential improve-

This work is financially supported by the Luxembourg National Research Fund (FNR) under the project SmartSpace (C21/IS/16193290).

ments offered by the proposed DL models.

II. System Model

In this paper, the emphasis is placed on satellites within GSO as the main system, and NGSO satellites, particularly in LEO orbits, are considered as possible sources of interference for GGS, as shown in Fig. 1.

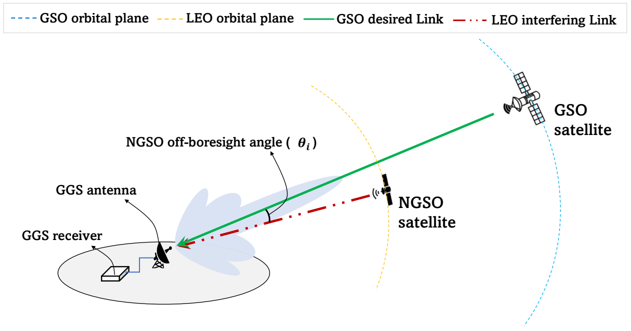


Fig. 1. NGSO-to-GSO interference scenario, where interference from NGSO is detected at GGS receiver unit

At a time instant ”t”, assuming that the impairments of up-down conversion are compensated ideally compensated, the desired received signal $y_x(t)$ atv the GGS is given as,

$$y_x(t) = x(t)\sqrt{\text{SNR}} + \zeta(t), \qquad (1)$$

where $x(t)$ is desired baseband signal, formed by modulated symbols obtained from a ModCod scheme following the DVB-S2X standard. Each symbol in $x(t)$ possesses an average power and span for a time interval T_x. The duration T_x is inversely proportional to the signal bandwidth B_x, as represented by the equation $T_x = \frac{1}{B_x}$. The signal $x(t)$ is shaped using a raised cosine filter, which has unit energy and a duration of T_x. The term $\zeta(t)$ denote the complex-valued additive white Gaussian noise (AWGN) with a zero mean, sustaining unit power across the bandwidth $2B_x$. Finally, SNR represents the received Signal-to-Noise Ratio (SNR) of the GSO signal, which can be calculated using the following equation,

$$\text{SNR[dB]} = P_{\text{EIR},x} + G_{r,\text{max}} - L_{\text{FS, x}} - L_{\text{A}} - \kappa - T_{\text{GS}} - B_{\text{x}}, \qquad (2)$$

where $P_{\text{EIR},x}$ denotes the GSO satellite equivalent isotropic radiated power (EIRP), which merges the power of transmission of the satellite antenna and its gain. The term $G_{r,\text{max}}$ denote the maximum receiving antenna gain of the GGS. The free space path loss (FSPL) at this link is represented by $L_{\text{FS, x}}$, while L_{A} denotes the additional link losses. κ denotes the Boltzmann constant,T_{GS} denotes the noise temperature of the receiver, while B_{x} refers to the GSO signal bandwidth.

Simultaneously, a LEO satellite may transmit an interfering signal towards GGS. Thus, the interference signal components $y_i(t)$ received at GGS can be expressed as follows,

$$y_i(t) = i(t)e^{j2\pi(f_{c,i}-f_{c,x})t}\sqrt{\text{INR}}, \qquad (3)$$

where $i(t)$, is the interfering baseband signal, consists of modulated symbols selected from a ModCod scheme in accordance with the DVB-S2X standard, can be same as GSO signal or different. This information is not known by the GGS. Each symbol in $i(t)$ possesses an average power and span for a duration of T_i. This time span is inversely proportional to the signal bandwidth B_i, thus $T_i = \frac{1}{B_i}$. The production of the $i(t)$ signal includes the utilization of a raised cosine filter with a duration of T_i and unit energy, in addition to a roll-off factor that varies between 0 and 1. The signal $i(t)$ undergoes down-conversion from its bandpass, utilizing its LEO carrier frequency $f_{c,i}$ and the GSO carrier frequency $f_{c,x}$. For the sake of simplicity, we assume that the bandwidths of both the desired and interfering signals completely overlap. Finally, in (3), INR represents the Interference-to-Noise ratio (INR) of LEO signal, which can be derived using the subsequent equation,

$$\text{INR[dB]} = P_{EIR,i} + G_{r,i}(\theta_i) - L_{\text{FS,i}} - \kappa - T_{\text{GS}} - B_{\text{x}}, \qquad (4)$$

where $P_{\text{EIR, i}}$ denotes the EIRP of LEO satellite. The term $G_{\text{r, i}}$ indicates the gain of the GGS receiving antenna with respect to the LEO satellite. θ_i is the off-boresight angle that determines the alignment of the GGS main beam lobe with the LEO communication link, see Fig. 1.

When there is interference from LEO, the combined signal received by the GGS, which includes the desired GSO signal components, interference from LEO signal components, and noise, can be mathematically represented as follows,

$$y(t) = x(t)\sqrt{\text{SNR}} + i(t)e^{j2\pi(f_{c,i}-f_{c,x})t}\sqrt{\text{INR}} + \zeta(t). \qquad (5)$$

At GGS, in the process of converting from analog to digital (A/D), the received signal is subjected to sampling at a frequency of $f_s = \frac{1}{T_s}$ Hz, where T_s denotes the sampling duration, the signal combined of M total number of I/Q samples. Eventually, we get a vector of complex-valued numbers $\mathbf{y}_x \in \mathbb{C}^M$ from (1) in case of GSO interference-free signals, or $\mathbf{y} \in \mathbb{C}^M$ as from (5) in the case of GSO signals with LEO interference.

III. Data Preparation

Assume we generated N data points of $\mathbf{y}_x$ or $\mathbf{y}$ signal samples. From now onwards, a data point in N will indexed as $\mathbf{y}_n \in \mathbb{C}^M$ where n = 1, 2,...,N. Later on, we utilize DL models based on unsupervised learning and need to train the models on interference-free data only, but the evaluation is based on interference and interference-free data. Thus, our datasets generation is composed of three parts:

1) Training datasets: We emulate an scenario with a single GSO satellite and a GGS who is pointing towards such GSO satellite and receiving the interference-free signal for a period of time $\mathcal{T}_{tr}$, and save N_{tr} data points using a small simulation step of $\Delta\tau_{tr}$. We repeat the simulation three times and vary the loss factor L_A which causes the adaptive coding and modulation scheme (ACM) to transmit using different ModCods. In particular, we consider three ModCods, that is, QPSK, 8PSK and 16APSK. The values of SNR and the signal samples are saved in three different sets, namely ($\mathbf{Y}_{tr,QPSK} \in \mathbb{C}^{N_{tr}\times M}$, $\mathbf{Y}_{tr,8PSK} \in \mathbb{C}^{N_{tr}\times M}$, and $\mathbf{Y}_{tr,16APSK} \in \mathbb{C}^{N_{tr}\times M}$). A fourth dataset is set by combining all three ModCods sets together: $[\mathbf{Y}_{tr,QPSK}, \mathbf{Y}_{tr,8PSK}, \mathbf{Y}_{tr,16PSK}]$, it can be denoted as $\mathbf{Y}_{tr} \in \mathbb{C}^{3N_{tr}\times M}$.

2) Validation datasets: Similar to the training datasets, we generate interference-free datasets for models validation, using $\mathcal{T}_{va}$, $\Delta\tau_{va}$, and generating N_{va} data points.

3) Testing datasets: For the testing datasets, we added LEO satellite to the simulation, and assumed it uses 16APSK ModCod all the time. For $\mathcal{T}_{ts}$, $\Delta\tau_{ts}$, here we collect N_{ts} data of SNR and INR values, then use them to generate both interference-free and interference signals samples for evaluating the models. Thus, the total generated testing data size is doubled. Specifically, we repeat the simulation three times[1], and store three different sets based on GSO ModCods, namely ($\mathbf{Y}_{ts,QPSK} \in \mathbb{C}^{2N_{ts}\times M}$, $\mathbf{Y}_{ts,8PSK} \in \mathbb{C}^{2N_{ts}\times M}$, and $\mathbf{Y}_{ts,16APSK} \in \mathbb{C}^{2N_{ts}\times M}$). Similar to the training and validation datasets, a fourth dataset is generated by combining all three ModCods matrices together, and it can be denoted as $\mathbf{Y}_{ts} \in \mathbb{C}^{6N_{tr}\times M}$. Finally, we manually added a data label column at the end of each dataset matrix, with an "0" label for interference-free data and an "1" label for interference data.

Given that machine learning algorithms do not support direct input data of complex numbers, we will explore the effectiveness of two different data representations for detecting interference:

1) **Received signals as amplitude values (1D-Data):** Generated by transforming the I/Q samples into a column vector of the signals magnitude values, each vector has a shape of M. The new datasets will be denoted by simply adding the letter $\mathcal{A}$ to the generated matrices, e.g., $\mathbf{Y}^{\mathcal{A}}_{tr}$. We will refer to this data representation as *Amplitude data*, and each data point can be expressed by the following equation,

$$y^{\mathcal{A}}_n = |\mathbf{y}_n| \in \mathbb{R}^M, \tag{6}$$

2) **Received signals as I/Q samples (2D-Data):** Generated by splitting I/Q samples into two columns (real and imaginary parts). From the DL perspective, all inputs are real numbers, composed as a matrix of shape (M, 2). The new datasets will be denoted by simply adding the letters $\mathcal{IQ}$ to the generated matrices, e.g., $\mathbf{Y}^{\mathcal{IQ}}_{tr}$. We will refer to this data representation as *I/Q data*, and each data point can be expressed by the following equation,

$$y^{\mathcal{IQ}}_n = [\mathrm{Re}\{\mathbf{y}_n\}; \mathrm{Im}\{\mathbf{y}_n\}] \in \mathbb{R}^{M\times 2}, \tag{7}$$

IV. Proposed Deep Learning Models

We propose DL models, based on CAE architecture, for interference detection tasks. This entails initially training the CAE using signal data devoid of interference. Subsequently, we aim to establish a reliable reconstruction error threshold. Ultimately, this threshold will be applied to signals, with or without interference. An interference is flagged when the detected error values exceed this predetermined threshold.

A. Notation and Assumptions

We denote a model input as y_n e.g., $y^{\mathcal{A}}_n$ or $y^{\mathcal{IQ}}_n$. Each input, before being fed to the model, undergoes an element-wise normalization, scaled between 0 and 1 based on the minimum value and the maximum value in the training dataset of interest.

A CAE model utilizes an encoder to compress any input y_n into a representation of lower-dimensional latent space. The decoder then decodes this lower-dimensional representation back to the original dimension, resulting in a reconstructed version of y_n. The objective of a CAE is to acquire a productive and significant representation of the input by reducing a loss metric between the initial input (y_n) and the reconstructed output ($\hat{y}_n$). The loss function employed to train and assess the performance of all models is the mean absolute error ($\mathcal{L}_{\mathrm{MAE}}$). The computation entails taking the average of the absolute differences between the predicted values and the actual values. The expected result is for CAE to successfully reconstruct data without any interference, resulting in a low $\mathcal{L}_{\mathrm{MAE}}$ value. However, if interference is encountered during the reconstruction process, a higher $\mathcal{L}_{\mathrm{MAE}}$ value will be observed. By establishing a threshold β for the $\mathcal{L}_{\mathrm{MAE}}$ values, it becomes feasible to distinguish between instances with no interference and instances with interference. Consequently, β can be considered as the interference score, such as,

$$d_n = \begin{cases} 1 \quad \textbf{Interfence}, & \text{if} \quad \mathcal{L}_{\mathrm{MAE}} > \beta \\ 0 \quad \textbf{Interfence-free}, & \text{otherwise} \end{cases} \tag{8}$$

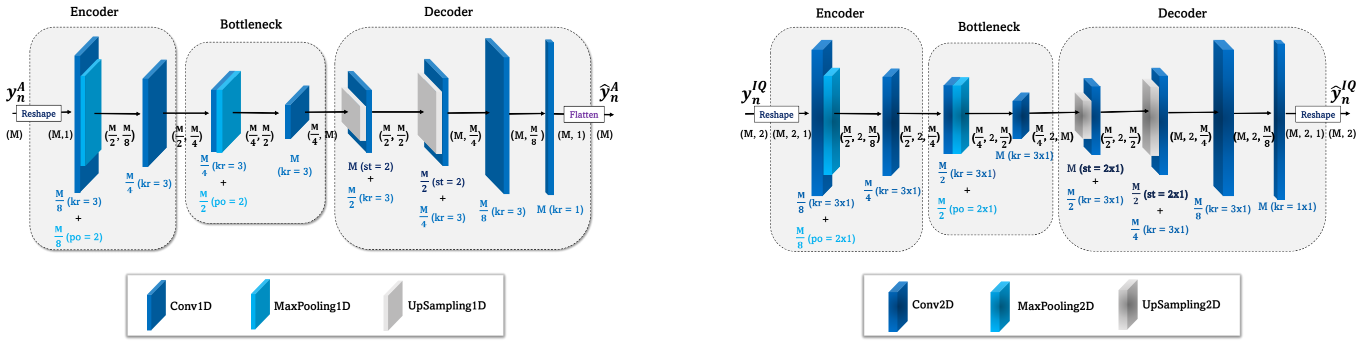
where d_n denotes the decision made by the model for the input y_n after going through the thresholding process.

B. Models Details

We present two separate CAE models, one for each data representations. Both models has equal complexity, which is often associated with the number of parameters (weights and biases) it has to learn. We chose the final dimensions and hyperparameters after preliminary testing experience.

1) 1D Convolutional Autoencoder Model (CAE2D): Fig. 2a detailed the CAE1D model architecture. Which dedicated to process 1D data (i.e., *Amplitude data*). In the encoder, the input is reshaped into a different dimension to prepare it for convolution. In this case, the input is a 2D array that is reshaped into a 3D tensor suitable for 1D convolution (batch, features, filters). M refers to the number of features in the input. Conv1D Layer performs 1D convolution on the input data, and $kr = 3$ specifies the kernel size of the convolution filter.

[1] Notice here all three times, INR data will have the exact values because no additional losses are added to (4), and this is intentional for the purpose of comparing the models detection performance in similar interference conditions.


Encoder
Bottleneck
Decoder
y_n^A
Reshape
(M)
(M,1)
$(\frac{M}{2}, \frac{M}{8})$
$(\frac{M}{2}, \frac{M}{4})$
$(\frac{M}{4}, \frac{M}{2})$
$(\frac{M}{4}, M)$
$(\frac{M}{2}, \frac{M}{2})$
$(M, \frac{M}{4})$
$(M, \frac{M}{8})$
(M, 1)
(M)
Flatten
$\hat{y}_n^A$
$\frac{M}{8}$ (kr = 3) + $\frac{M}{8}$ (po = 2)
$\frac{M}{4}$ (kr = 3)
$\frac{M}{4}$ (kr = 3) + $\frac{M}{2}$ (po = 2)
M (kr = 3)
M (st = 2) + $\frac{M}{2}$ (kr = 3)
$\frac{M}{2}$ (st = 2) + $\frac{M}{4}$ (kr = 3)
$\frac{M}{8}$ (kr = 3)
M (kr = 1)
Conv1D
MaxPooling1D
UpSampling1D
Encoder
Bottleneck
Decoder
y_n^{IQ}
Reshape
(M, 2)
(M, 2, 1)
$(\frac{M}{2}, 2, \frac{M}{8})$
$(\frac{M}{2}, 2, \frac{M}{4})$
$(\frac{M}{4}, 2, \frac{M}{2})$
$(\frac{M}{4}, 2, M)$
$(\frac{M}{2}, 2, \frac{M}{2})$
$(M, 2, \frac{M}{4})$
$(M, 2, \frac{M}{8})$
(M, 2, 1)
(M, 2)
Reshape
$\hat{y}_n^{IQ}$
$\frac{M}{8}$ (kr = 3x1) + $\frac{M}{8}$ (po = 2x1)
$\frac{M}{4}$ (kr = 3x1)
$\frac{M}{2}$ (kr = 3x1) + $\frac{M}{2}$ (po = 2x1)
M (kr = 3x1)
M (st = 2x1) + $\frac{M}{2}$ (kr = 3x1)
$\frac{M}{2}$ (st = 2x1) + $\frac{M}{4}$ (kr = 3x1)
$\frac{M}{8}$ (kr = 3x1)
M (kr = 1x1)
Conv2D
MaxPooling2D
UpSampling2D

(a) CAE1D Model

(b) CAE2D Model

Fig. 2. Proposed Deep Learning Models

MaxPooling1D layer performs downsampling by taking the maximum value over a pool size, reducing the dimensionality of the data, $po = 2$ indicates the pooling size. The Bottleneck layers continue reducing the features dimension to the latent space representation of size $\frac{M}{4}$.

In the Decoder, an UpSampling1D layers performs the opposite operation of max pooling it increases the dimensionality of the data by repeating the values, effectively undoing the downsampling from the encoder. Finally, a Conv1D layer that has M filters and a kernel size of 1 generates a solitary feature map. This feature map is subsequently flattened, and produce the reconstructed output $\hat{y}_n$, which has a size of M .

2) 2D Convolutional Autoencoder Model (CAE2D): Fig. 2a illustrates a CAE2D model architecture. The primary difference from CAE1D is that the CAE2D model will only process on 2D data (i.e., *I/Q data*), which is reflected in the use of 2D convolutional, pooling, and upsampling layers. In this case, the input is a 3D array that is reshaped to a 4D tensor, suitable for 2D convolution (batch, features, height, filters). The reshaping of the input and output to accommodate the two-dimensional nature of the data.

C. Performance Metrics

The effectiveness of these models in identifying GSO interference-free instances (negative classes), and NGSO interference instances (positive classes) at GGS is evaluated using the following key performance indicators [6], 1) Accuracy score evaluates the percentage of accurate predictions made by the model out of all the predictions, 2) Precision score refers to the accuracy of the model in correctly detecting instances of interference, 3) Recall score indicates the model sensitivity to instances of interference, in our scenario it also indicates, the probability of detection, and the true positives rate. 4) $\mathrm{F1-Score}$ is the harmonic mean of precision and recall, 5) AUC score is a scalar value indicates the area under the receiver operating characteristic curve (ROC) curve, this value quantifies the model capability to differentiate between NGSO interference instances and GSO interference-free instances.

V. Experimental Results

A. Data Generation

The NGSO-to-GSO satellites simulation and SNR and INR values calculations conducted using the same approach explained in details in [5]. The simulation UTC startTime is 01-Aug-2023 06:07:00. The orbital movement of the chosen GSO and LEO satellites is computed using the built-in functions in MATLAB. These functions give approximations of the latitude, longitude, and altitude at different points in time based on the satellites' trajectory. The link budget parameters are shown in Table I.

TABLE I
Simulation and Link Budget Parameters

Parameters	Values
$\mathcal{T}_{tr}, \mathcal{T}_{va}, \mathcal{T}_{ts}$	120 seconds
$\Delta\tau_{tr}, \Delta\tau_{va}, \Delta\tau_{ts}$	50, 16 & 16 ms
N_{tr}, N_{va}, N_{ts}	7200, 2400 & 2400
$P_{EIR,x}$	52.2 dBW
B_x	50 MHz
$f_{c,x}$	11.750 GHz
L_{A}	0 to 9 dB
$P_{\mathrm{EIR},i}$	44.7 dBW
B_i	50 MHz
$f_{c,i}$	11.750 GHz
T_{GS}	250° K
$G_{r,\max}$	42.2 dBi

The values of $L_{\mathrm{FS},i}$ and $G_{r,i}(\theta_i)$, from (4), vary with the changing positions of the STARLINK satellite. With this setup, we generated GSO SNR values for each ModCod as follows, 1) QPSK ModCod for SNR between [2 - 5.9] dB, 1) 8PSK ModCod for SNR between [6 - 9.9] dB, and 3) 16APSK ModCod for SNR between [10 - 14.9] dB. For the testing data, the values of INR range from [-24.4 to 33.3] dB for all GSO ModCods scenarios, as shown in Fig. 3. Furthermore, the roll-off factor of the raised cosine filter was set to 0.25 and the sampling frequency f_s = 100 MHz, which generated signals with a total of M = 800 samples.

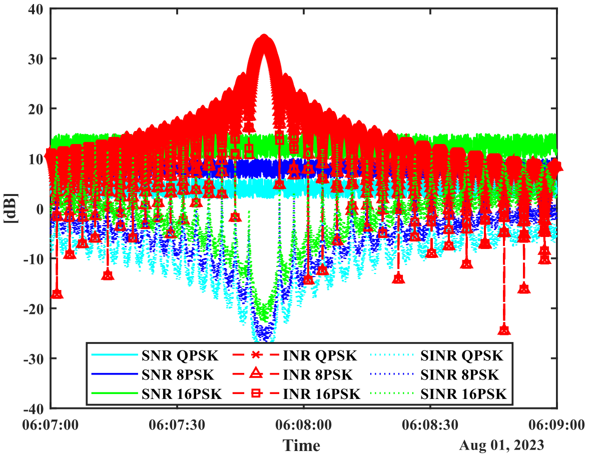


Fig. 3. SNR and INR values per GSO ModCod

B. Training the autoencoder model

For one data representation (*Amplitude data*, or *I/Q data*), we generated a dataset matrix for each GSO ModCod separately, then an additional dataset matrix combine all matrices is generated as explained in Section III. In this part, we investigate the model performance using two types of approaches for training.

1) **Single Model approach (SM):** where we train and evaluate one model to detect interference in all GSO ModCods, i.e. using the combined GSO ModCods datasets.
2) **Multi-Models approach (MMs):** where we train and evaluate the model to detect interference on specific GSO ModCods, i.e., using the separated GSO ModCods datasets. Finally, we will have three trained models that collaboratively detect interference.

The models proposed in Section IV are constructed using the Python with the assistance of the Tensorflow and Keras frameworks. MAE is chosen as the primary loss function for all models, and the optimizer utilized is Adam. The weights parameters are updated through the use of training datasets, and the validation datasets are employed to validate the results. All the models underwent training for a total of 80 epochs, using a mini-batch size of 64. Fig. 4 displays an input data, the reconstructed output, and the reconstruction error regions, obtained from the CAE1D using the SM approach, after reverses normalization. The presence of interference in the data indicates a higher reconstruction error (failed reconstructions). This outcome is anticipated because the models were trained using interference-free data, making it more challenging to reconstruct interference data.

We suggests utilizing an initial threshold value β_0 for each model and dataset to differentiate between interference-free and interference data. This value can be obtained from a combination of mean and standard deviation values of the MAE loss values in the training dataset $\mathcal{L}_{\text{MAE,Train}}$. This threshold value ensure a satisfactory performance in detecting interference. However, in this paper, we have expanded the definition of the threshold to include an improved value (β) by analyzing interference segments (assuming they are accessible). Specifically, we identify the threshold value that gives the highest AUC score by utilizing the testing dataset. For the sake of the space, we use β only, although it can be extended to discuss the performance of the β_0 and β thresholds.

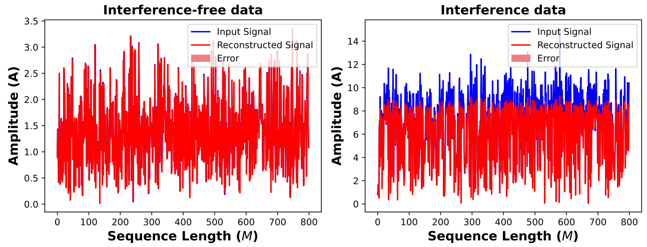


Fig. 4. Input and reconstructed output for an interference-free (left) and interference data (right)

C. Models evaluation results

The models performance is evaluated by comparing them with a conventional ED method. In classical detection theory, an ED is considered the optimal likelihood-ratio test for detecting stochastic signals in environments with white Gaussian noise [7]. The energy E_n, which is the square of the magnitude, is determined by summing the squared amplitudes of the samples of the signal $\mathbf{y}_n$. An energy threshold β_E is established to decide if signal y_n contains NGSO interference. The threshold is constant and is determined by the processed test datasets using the optimal AUC score approach.

Our proposed CAE1D and CAE2D models, each has 76.21% fewer parameters than the CAE model in [4], indicating a simpler structure that would typically be easier to train and less prone to overfitting. Thus, we also includes a performance comparison of our results with the CAE model approach in [4] to show the superiority of our models.

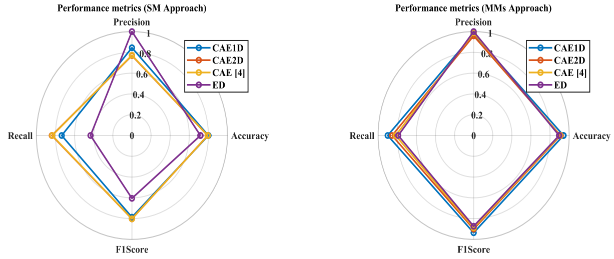


Fig. 5. Models performance metrics for The SM approach (Left), and the MMs approach (Right)

Fig. 5 shows radar sharts with performance metrics results for the proposed models in comparison to CAE [4], and the ED method, using SM and MMs approaches. In particular, it highlights that all DL models show a very similar performance,

TABLE II
PERFORMANCE METRICS SUMMARY OF THE PROPOSED MODELS

Model	Input Data	Train Based	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC (%)
CAE1D	Amplitude	SM	79.99	84.44	73.51	78.60	79.99
		MMs	**94.01**	97.75	**89.94**	**93.57**	**94.01**
CAE2D	I/Q	SM	79.30	76.87	83.82	80.19	79.30
		MMs	90.92	95.88	85.29	90.07	90.92
CAE [4]	I/Q	SM	79.39	77.20	83.42	80.19	79.39
		MMs	90.33	98.82	81.56	88.77	90.33
ED	Energy	SM	71.67	**100.0**	43.35	60.48	71.67
		MMs	89.47	**100.0**	78.93	87.33	89.47

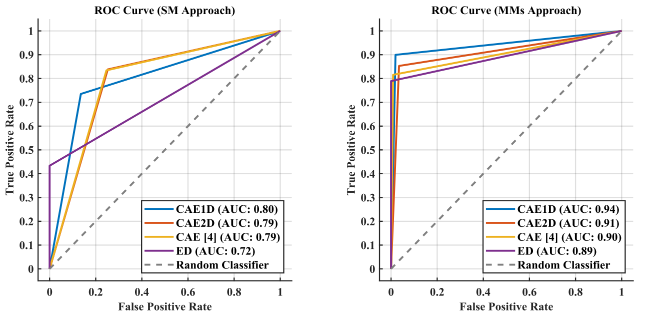


Fig. 6. ROC curves per model for the SM approach (Left), and the MMs approach (Right)

with the lines overlapping closely, which suggests that all models perform similarly across the metrics when trained with either the SM or MMs approaches in comparison to the ED method. Fig. 6 shows ROC curves, comparing the same models as previously discussed CAE1D, CAE2D, CAE [4], and ED, using the SM and MMs training approaches. The CAE1D model has an AUC of 0.80 under the SM approach and improves significantly to 0.94 under the MMs approach. While CAE [4] and CAE2D each have a AUC of 0.79 for the SM approach, which increases to 0.90 and 0.91, respectively, with the MMs approach. On the other hand, the ED model exhibits an AUC of 0.72 with the SM approach and a higher AUC of 0.89 with the MMs approach. The consistent improvement across all models when using the MMs training approach suggests that this approach may be better suited for training models.

Table II, summarizes the performance metrics of the proposed models CAE1D, CAE2D, CAE [4], and ED in both training approaches. All metrics are expressed as percentages. For CAE1D (which processes *Amplitude data*) and CAE2D (which processes *I/Q data*), precision is particularly high with the MMs approach, suggestion that when they predict an instance of interference, they are highly likely to be correct. CAE [4] has a better precision score under MMs approach, but low scores otherwise. Finally, the ED model has a perfect precision score under both SM and MMs training, but with low recall. The precision and recall are the only metrics in which CAE2D and *I/Q data* perform better than CAE1D when both are on SM training. Overall, in terms of detecting interference and minimizing false alarms, the CAE1D model trained with MMs approach demonstrates superior performance compared to other DL models and the ED method.

VI. CONCLUSION

The proposed CAE1D (and *Amplitude data*), when trained with the MMs approach, showed remarkable improvement across all performance metrics such as accuracy, precision, recall, F1-Score, and AUC, compared to the CAE2D model (and *I/Q data*) as well as the traditional ED method. This improvement underscores the effectiveness of deep learning in complex interference scenarios and suggests a direction for future research and development in satellite communication security and reliability. The University of Luxembourg HPC facilities were utilized for conducting the experiments [8]. The data utilized in this study will be accessible to the public through the Smart-Space project website[2].

REFERENCES

[1] S. K. Sharma, S. Chatzinotas, and B. Ottersten, “In-line interference mitigation techniques for spectral coexistence of GEO and NGEO satellites,” *International Journal of Satellite Communications and Networking*, vol. 34, no. 1, pp. 11–39, Sep. 2014.

[2] C. Braun, A. M. Voicu, L. Simić, and P. Mähönen, “Should we worry about interference in emerging dense NGSO satellite constellations?” in *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2019, pp. 1–10.

[3] L. Pellaco, N. Singh, and J. Jalden, “Spectrum prediction and interference detection for satellite communications [international communications satellite systems conference],” in *Advances in Communications Satellite Systems. 37th International Communications Satellite Systems Conference (ICSSC-2019)*. Institution of Engineering and Technology, 2019.

[4] M. A. Vazquez, P. Henarejos, I. Pappalardo, E. Grechi, J. Fort, J. C. Gil, and R. M. Lancellotti, “Machine learning for satellite communications operations,” *IEEE Communications Magazine*, vol. 59, no. 2, pp. 22–27, Feb. 2021.

[5] A. Saifaldawla, F. G. Ortiz-Gomez, E. Lagunas, S. Daoud, and S. Chatzinotas, “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, pp. 1–7.

[6] A. Tharwat, “Classification assessment methods,” *Applied Computing and Informatics*, vol. 17, no. 1, p. 168–192, Jul. 2020. [Online]. Available: http://dx.doi.org/10.1016/j.aci.2018.08.003

[7] S. Kay, *Fundamentals of Statistical Signal Processing: Detection theory*, ser. Fundamentals of Statistical Si. Prentice-Hall PTR, 1998.

[8] S. Varrette, H. Cartiaux, S. Peter, E. Kieffer, T. Valette, and A. Olloh, “Management of an Academic HPC & Research Computing Facility: The ULHPC Experience 2.0,” in *Proc. of the 6th ACM High Performance Computing and Cluster Technologies Conf. (HPCCT 2022)*. Fuzhou, China: Association for Computing Machinery (ACM), July 2022.

[2] https://fnr-smartspace-project.uni.lu