TUM

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY - INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Bachelor's Thesis in Informatics

Deep Learning Based Space Debris Classification in Low Earth Orbit Using Space-Borne Radar Simulations

Oscar Breiner

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY - INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Bachelor's Thesis in Informatics

Deep Learning Based Space Debris Classification in Low Earth Orbit Using Space-Borne Radar Simulations

Deep Learning Basierte Klassifizierung von Weltraumschrott in Niedriger Erdumlaufbahn Mittels Simulationsgestützter Radarmessungen aus dem Weltraum

Author:	Oscar Breiner
Supervisor:	Prof. Dr. Stephan Günnemann
Advisor:	Marten Lienen
Submission Date:	15.03.2024

I confirm that this bachlor's thesis is my own work and I have documented all sources and material used.

Munich, 15.03.2024   Oscar Breiner

Abstract

This thesis addresses the challenge of classifying centimeter-sized space debris in Low Earth Orbit (LEO) using radar technology and deep learning techniques, given the increasing threat they pose to active space infrastructures. We propose deploying space-borne radar systems to overcome the limitations of conventional space debris observation methods in detecting small debris particles. Therefore, we developed a simulation pipeline designed to mimic a monostatic radar system. This system emits high-frequency radar radiation at 94 GHz and captures the reflections from rotating debris. Since this operation is deployed within the optical scattering region, it enables us to extract spatial characteristics of the target object from the captured signal. The target objects are composed of fully conducting materials and are categorized into four distinct classes: cones, cylinders, spheres, and circular plates, each containing instances in varying shapes and sizes.

The resulting dataset of captured signals is used to evaluate the application of deep learning based space debris classification. The examined deep learning architectures include Multilayer Perceptron (MLP), Long Short-Term Memory (LSTM), Residual Network (ResNet), and Transformer models equipped with either Sparse-Attention or Full-Attention mechanisms.

Experimentation highlighted the impact of signal processing. Decibel transformation leads in many cases to improved accuracy, better generalization, and faster convergence. However, bidirectional LSTM and Transformers are exceptions to this trend, demonstrating the capability to effectively process either raw signal samples or decibel signals.

Through extensive experimentation and hyperparameter tuning, we achieved accuracy results in classification close to 95%. This threshold is linked to similar radar cross-section (RCS) signatures from certain observational angles of tall cones and cylinders.

The simulation data also factors in real-world challenges in the form of additive Gaussian noise and RCS signature fluctuations due to object vibrations. For further robustness evaluations, we tested for radar-specific noise on pretrained models by distorting the test set with various types of signal occlusion, clutter, and sensor saturation. The results underscore the effectiveness of applying decibel transformations before feature extraction. Models trained on decibel signals appear to benefit from the noise reduction capabilities of the logarithmic scale, improving clarity in radar distortions like clutter and saturation. Yet again, Full-Attention Transformers trained on raw signals defy this trend of better decibel efficacy, showcasing unmatched resistance to occlusion and various frequency-based clutter scenarios, outperforming every other examined model.

Overall, this research thesis highlights the potential of deep learning for classifying radar targets, improving the surveillance of space debris and improving the safety of space operations. It showcases the application of space-borne radar systems and emphasizes the capabilities of deep learning models in processing complex radar signals.

Contents

Contents

1 Introduction

Over the past decades, space debris has become an increasing threat to the survival of space infrastructures. Inactive satellites, fragmented parts of rockets, and pieces left over from space missions have led to millions of pieces of space junk orbiting Earth, many of which remain untracked and uncatalogued. Therefore, the current numbers of debris objects in orbit are estimated based on statistical models provided by ESA's MASTER-8 population function [1].

Most space debris is located in Low Earth Orbit (LEO) at altitudes between 800 km and 1400 km, where the average impact speed reaches up to $10 - 15$ km/s. At these velocities, even centimeter-sized particles may cause serious damage to operational spacecraft upon impact [2].

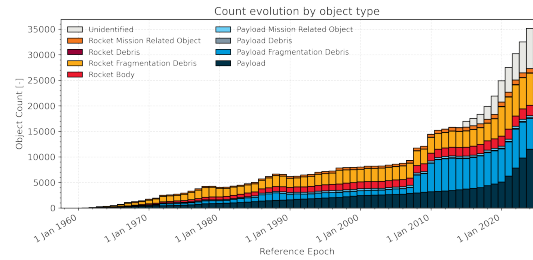| Size of Space Debris	Quantity
Greater than 10 cm	36,500
Between 1 cm and 10 cm	1,000,000
Between 1 mm and 1 cm	130,000,000

Figure 1.1: Quantities of space debris objects estimated by ESA in 2021 [1]

In response, efforts have been made to detect, track, and log space debris using optical sensors and ground-based radars [3], [4], [5]. Although these systems can track objects larger than 10 cm, their ability to detect smaller particles remains limited [4].

However, space-borne radar systems might offer a solution to the challenge of monitoring small debris sizes. Being closer to their targets and not being affected by atmospheric noise, it allows the usage of higher frequency bands. Furthermore, they require less emission power, enhancing their efficiency and effectiveness in tracking space debris [6].

To reduce the risk of space debris in LEO, the classification of debris is just as important as detection. By assessing the shape of objects, one can determine material properties and conduct risk assessments. Furthermore, classification aids in reidentification for debris surveillance and can help pinpoint the origin of specific spacecraft fragments to avoid future debris creation.

In the past, stochastic models, such as NASA's size estimation model (SEM) [7], have been the cornerstone for evaluating space debris statistics. However, the rise of powerful GPUs in the last decade has made Deep learning an interesting alternative solution. Most Deep Learning models, tasked with classification, are deployed in a Supervised Learning setting,

requiring large amounts of labeled data. However, acquiring radar datasets poses a significant challenge, since most radar-related datasets are either classified by the military or inaccessible for public use.

To address this challenge, we employ a simulation by our own design to calculate the interaction between high-frequency radar radiation and space debris ranging from 1 to 10 cm. The target objects consist out of cones, cylinders, circular plates, and spheres in various shapes and are simulated to be in motion, following rotational patterns in two dimensions. To mimic real-world conditions, the simulation includes additive Gaussian noise and fluctuation to represent the vibration of the objects due to their motion. These fluctuations are calculated based on the stochastic approach of Swerling Model [8].

The resulting dataset consists out of observation windows of 501 time steps, representing the reflected radar signal to the source of emission. This process is modeled on the principles of monostatic radar systems and considers the radar cross-section (RCS) of each object type. A detailed explanation of the simulation pipeline is provided in Chapter 2.

The simulated dataset enables us to train classification models in a supervised learning setting. The objective of this thesis is to evaluate the effectiveness of various deep learning architectures in the classification of radar-detected space debris. However, this classification task encounters challenges due to unknown motion patterns and the spatial relationship between the target and the radar emitter. This leads to classification difficulties for specific circumstances, especially for certain cone and cylinder instances based on their shape and their orientation during observation. Despite these challenges, the classification task achieves impressive accuracy levels of up to 95%, thanks to domain-specific data preprocessing techniques and extensive training, including hyperparameter tuning. A detailed report on these experiments can be found in Chapter 3.

To mimic the unpredictable conditions of space and to ensure that the trained models could withstand real-world operational stress, further testing is crucial. Therefore, we introduce radar-specific noise factors such as clustering, random dropouts (data loss), and saturation to examine robustness in Chapter 4.

Overall, this thesis demonstrates the potential of deep learning applications in space debris classification, paving the way for safer space operations.

2 Dataset Simulation using MATLAB

The overall objective of this thesis is to examine the potential of deep learning for the classification of space debris. To achieve this, we simulate space debris as 3D models categorized into four geometric shapes: spheres, cylinders, circular plates, and cones. These objects vary in shape and size to represent the diverse array of debris encountered in space.

The resulting dataset is composed of time series data, which captures the radar signal reflections from the target object back to the emission source. This dataset is essential for training and evaluating the effectiveness of deep learning algorithms in classifying space debris based on their geometric properties.

2.1 Radar System and Related Work

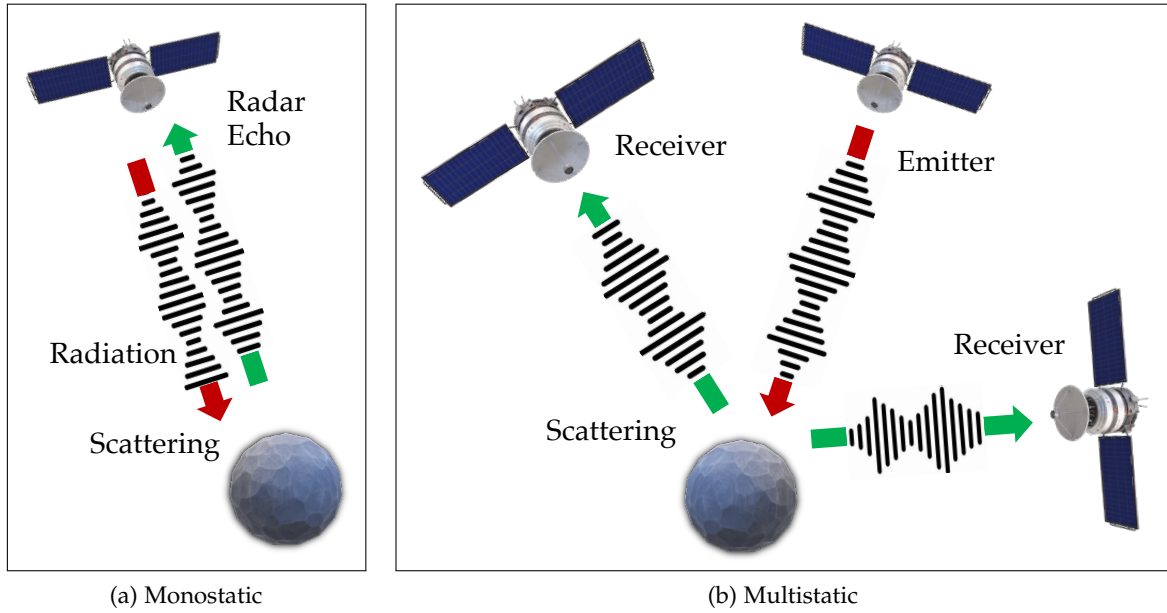(a) Monostatic   (b) Multistatic

Figure 2.1: Possible radar settings for space-borne radar systems

In the design of our radar setup, we differentiate between multistatic and monostatic radar systems, as depicted in Fig. 2.1. Multistatic radar systems employ a network of spatially separated receivers and transmitters to capture reflected radar signals of a target across multiple channels. These systems have shown promising results with the use of

Convolutional Neural Networks (CNNs) for object classification, where the addition of radar channels has been linked to improved accuracy [9]. In contrast, monostatic radar systems deploy a single co-located receiver– transmitter setup, thereby facing the challenge of relying on a single data stream. This limitation might restrict the diversity of features available for analysis. Despite this challenge, monostatic systems have proven effective in recent deep learning driven classification endeavors, utilizing both CNNs and Recurrent Neural Networks (RNNs) [10][11].

Given the cost-effective libraries available for monostatic radar configurations within MATLAB's Radar Toolbox [12], we have chosen to implement a monostatic setup, drawing inspiration from the research conducted by the Institution of Engineering and Technology (IET) [10]. This decision is driven by the advantages of simpler calibration, reduced data fusion complexity, and lower deployment costs for the simulation pipeline. Additionally, should this project concept be applied in a real-world scenario, it would likely begin with a single satellite or payload due to the steep financial implications of launching and maintaining multiple satellites.
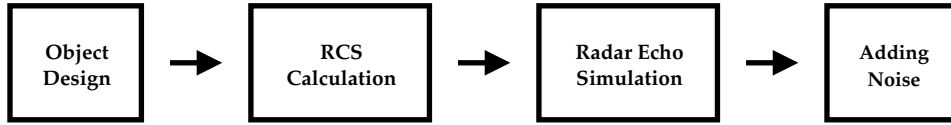
2.2 Simulation Configuration and Workflow

Figure 2.2: Structure of the simulation pipeline

The objective of this simulation is to accurately capture the spatial characteristics of space debris objects in Low Earth Orbit (LEO), specifically those ranging in size from 1cm to 10cm. To achieve this, we are implementing a monostatic radar system integrated into satellite platforms.

This idea of employing space-borne radar for space debris observation is inspired by the research conducted by the team of Ramírez-Torres [6]. This line of research was initiated in response to the limitations of radar ground stations and space-borne optical sensors, which, while effective at detecting objects larger than 10cm, struggle to monitor debris in the critical 1-10cm range. This size range is particularly concerning since centimeter-sized particles are still able to cause serious damage to operational space crafts with an average impact speed of 10-15 km/s.

Therefore, space-borne radars emerge as a promising solution for pinpointing these smaller debris fragments. Being closer to their targets and not being affected by atmospheric noise, it allows the usage of higher frequency bands. Furthermore, they require less emission power, enhancing their efficiency and effectiveness in tracking space debris [6].

2.2.1 Selecting the Operation Frequency

The effectiveness of our radar configuration in identifying the spatial features of space debris is determined by the chosen operation frequency. To optimize this selection process, we can review existing research within the field of space exploration.

An interesting initiative within this domain, was the planned installation of a space-borne radar system at the International Space Station, designed to detect small space debris fragments ranging from 1cm to 10cm [13]. Although this operation, known as "Project Warden", was never realized, its core idea can be integrated into our simulation efforts.

Drawing inspiration from the Warden Project, we choose an operating frequency of 94GHz for our simulation efforts. This frequency was specifically selected for its capability to detect centimeter-sized debris objects, which places this operation in the optical spectrum! Why is this important?

The success of radar classification through the radar-object scattering mechanisms is determined by the object size $\mathbf{L}$ in relation to the radar wavelength λ. In general, one can distinguish between three scattering regimes: Rayleigh, resonant, and optical (high frequency) region [14]. Each regime has a distinct implication on an object's radar signature, which is explained in Table 2.1.

Type	Characteristics	Explanation
Rayleigh	$\lambda \gg L$	The object scatters incoming radar waves in various directions, instead of back to the source. This leads to a weak radar signature not indicating any properties of the object, since the object is acting as a reradiating dipole.
Resonant	$\lambda \approx L$	The radar echo is highly sensitive to the object's orientation due to multiple scattering and reflection points. This mixture of reflections accumulates into a complex return signal, making interpretation difficult.
Optical	$\lambda \ll L$	The radar wavelength is small enough to interact with the smallest features of the object's surface, making the scattering highly dependent on its geometry and texture. This scenario provides a complex but informative signal for analysis.

Table 2.1: Scattering types and their characteristics [14]

Choosing an operative frequency of 94 GHz guarantees that objects of $1-10$ cm diameter are within the optical reflection region. In this regime, the radar cross-section (RCS) reaches its peak and can be closely approximated by the object's frontal projection area. This strategic

choice of frequency thereby maximizes the system's sensitivity to variations in target size and shape, enhancing its capabilities for classification tasks.

In addition, within the optical scattering domain, the RCS becomes less sensitive to frequency variations, reducing the necessity for a wide frequency range. Therefore, the utilization of a single, well-chosen frequency is sufficient without compromising the geometric characteristics of the radar image.

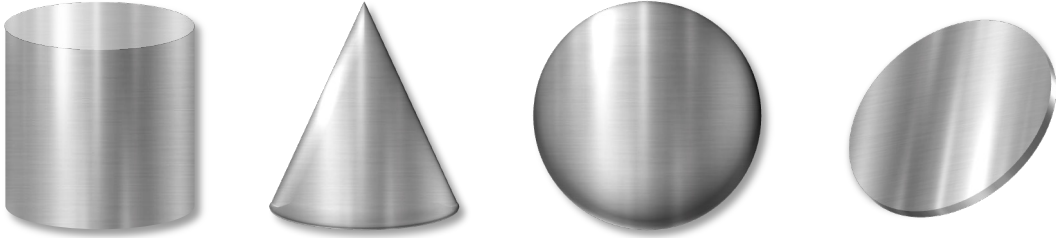### 2.2.2 Designing the Geometric Objects

Figure 2.3: Example illustration of the 4 geometric object types

The radar signature of an object is influenced by multiple factors including the radar wavelength and sampling rate. However, object-specific properties are also relevant such as surface material, shape, and motion. In this simulation, the objects under investigation are convex shapes made from a uniform material that behaves as a perfect electrical conductor. This setting emphasizes the object's shape and surface for determining its radar signature, aligning with the sensitivity of optical high-frequency regions to an object's geometry.

The set of target objects includes circular cones, circular cylinders, spheres, and circular plates.

For each class, **32 different instances** were designed with varying spatial parameters. Varying sizes across all three dimensions do not only lead to various scaled versions of the same shape but also introduce shape differences within the same object type, enriching the dataset for more comprehensive classification experiments.

Object Type	Geometrical Parameter	Dimensional Size Range
Spheres	Radius r	[0.005, 0.05] m
Cylinders	Height h	[0.01, 0.1] m
	Radius r	[0.01, h] m
Circular Plates	Radius r	[0.005, 0.05] m
Cones	Height h	[0.01, 0.1] m
	Radius r	[0.005, h] m

Table 2.2: Geometric object design parameters

2.2.3 RCS Calculation

The radar cross-section (RCS) of an object describes the amount of energy that is reflected directly back towards the source of emission. As an object changes its orientation in space relative to the radar beam, the RCS changes as well. This spatial orientation is described by the two aspect angles azimuth and elevation. Given the symmetrical nature of our object instances, the RCS map of each object can be described by an azimuth sweep from -180° to 180° and an elevation sweep from -90° to 90°.

$$RCS(\phi, \theta) : [-\pi, \pi] \times [-\pi/2, \pi/2] \rightarrow \mathbb{R} = f(\phi, \theta) \tag{2.1}$$

where:

- ϕ represents the azimuth angle, measured in the horizontal plane,

- θ represents the elevation angle, measured in the vertical plane,

- and $f(\phi, \theta)$ symbolizes the reflective properties of the object in question.

Moving forward, the RCS signature of each class instance is captured in form of a two-dimensional matrix. This matrix maps out an object's reflectivity across the dimensions azimuth and elevation, employing a high-resolution sampling rate of 0.1 degrees.

The illustrations in Section 2.2.3 showcase the RCS signatures of the four object classes, highlighting variations within each category. In the case of cones and cylinders, we differentiate between wide objects ($r \geq h$) and tall objects ($r \ll h$). For the circular plate and the sphere class, we examine the impact of size by comparing the RCS signatures of objects with small and large radii. These examples illustrate how variations in shape and size influence RCS signatures, providing an early insight into possible implications for the classification task ahead. For instance, while the RCS signature of cones changes quite drastically, cylinders almost remain the same. Spheres present a unique case, as they exhibit constant reflectivity regardless of their orientation.
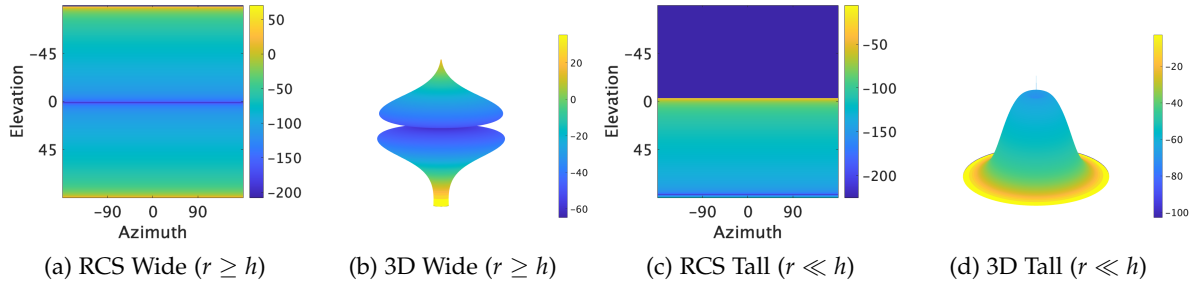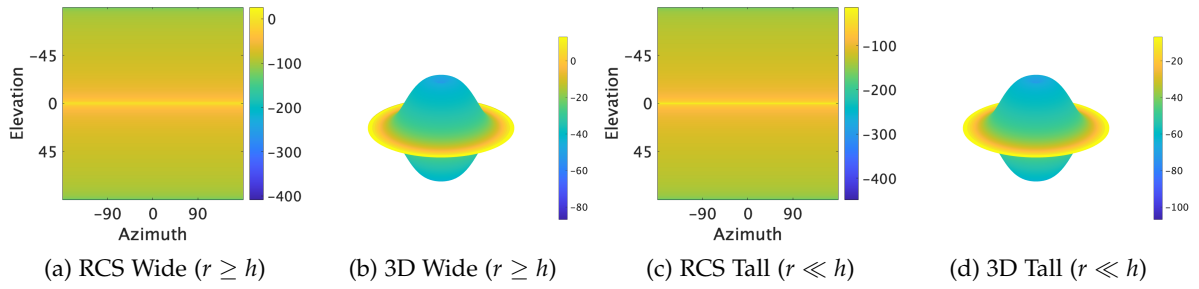
(a) RCS Wide ($r \geq h$)   (b) 3D Wide ($r \geq h$)   (c) RCS Tall ($r \ll h$)   (d) 3D Tall ($r \ll h$)

Figure 2.4: Example cone RCS signatures

(a) RCS Wide ($r \geq h$)   (b) 3D Wide ($r \geq h$)   (c) RCS Tall ($r \ll h$)   (d) 3D Tall ($r \ll h$)

Figure 2.5: Example cylinder RCS signatures

(a) RCS ($r = 0.005m$)   (b) 3D ($r = 0.005m$)   (c) RCS ($r = 0.05m$)   (d) 3D ($r = 0.05m$)
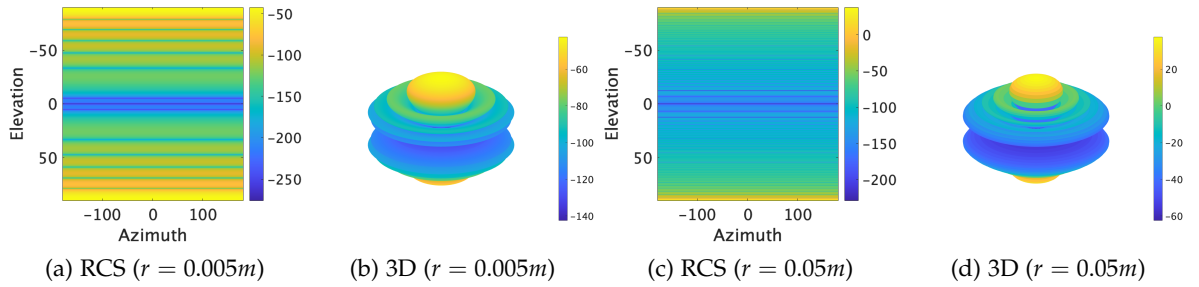
Figure 2.6: Example plate RCS signatures

(a) RCS ($r = 0.005m$)   (b) 3D ($r = 0.005m$)   (c) RCS ($r = 0.05m$)   (d) 3D ($r = 0.05m$)
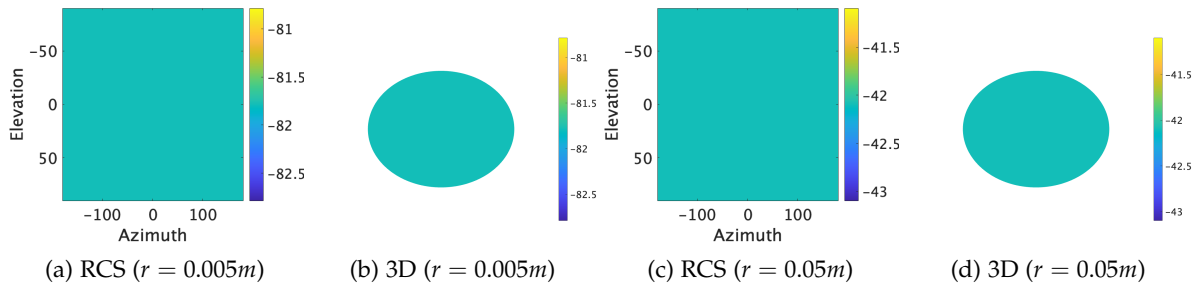
Figure 2.7: Example sphere RCS signatures

2.2.4 Simulating the Radar Echo

The simulation setup described is designed to model the reflection of radar signals by space debris without simulating the radar antenna itself. This is achieved by directly calculating the reflected radar echo, considering the object's rotation and the incoming direction of the radar signal, utilizing the RCS Map created earlier. If the RCS Map's resolution is insufficient, interpolation is employed to enhance accuracy.

To generate a time series signal, the setup employs a linear motion path to simulate an object's rotation. This trajectory can be described as a movement along a spherical surface by methodically sampling across the azimuth and elevation spectrum. To simulate various velocities, different sampling rates are introduced. Furthermore, the initial object orientation at the start of each motion path is selected at random, providing a diverse range of initial conditions.

$$\mathbf{f}(t) = \begin{pmatrix} \phi(t) \\ \theta(t) \end{pmatrix} = \begin{pmatrix} \phi_0 \\ \theta_0 \end{pmatrix} + t \cdot \begin{pmatrix} v_\phi \\ v_\theta \end{pmatrix} \tag{2.2}$$

where:

- $\mathbf{f}(t)$ describes the orientation of an object at time t, comprising azimuth and elevation components.

- $\begin{bmatrix} \phi_0 \\ \theta_0 \end{bmatrix}$ represents the initial orientation of the object at the start of the simulation ($t = 0$).

- $\begin{bmatrix} v_\phi \\ v_\theta \end{bmatrix}$ denotes the velocity components in terms of degrees per time step.

- t is the current time step.

Each simulation scenario is calculated across 501 time steps, limiting the observation window for each motion path.

It is important to add that moving objects undergo vibrations, causing the RCS map to change over time due to variations. To accommodate these changes, a statistical model known as the Swerling Model is applied. Specifically, for objects that exhibit rapid fluctuations, the Swerling Model 2 is utilized, which helps in accurately simulating the dynamic nature of RCS for moving objects.

Furthermore, it is important to add that all objects in motion experience vibration. As a result, the RCS map of a moving object does not remain constant over time. To account for these fluctuations, we employ a robust statistical approach known as Swerling Model [8]. Specifically, we apply Swerling Model 2 to simulate rapid object fluctuations to ensure a precise representation of the changing RCS signature.

2.2.5 Adding Gaussian Noise

In Low Earth Orbit, one can encounter many different noise sources, including cosmic background radiation, interference, atmospheric noise, and noise specific to the antenna of

the radar setup.

To represent different sources of noise, we add additive Gaussian noise to the radar echo, by applying various signal-to-noise ratios (SNR) selected at random.

Additionally, there are specific radar-related noise sources, such as saturation and random information loss. However, these will not be included in the default dataset for supervised training and will only be introduced during robustness testing on pretrained models in Chapter 4.

2.2.6 Resulting Dataset

The comprehensive simulation pipeline generates 128,000 one-dimensional monostatic RCS time series data samples. The parameters used in this simulation are summarized in Table 2.3. The raw signal amplitudes within each data sample are displaying voltage changes over time, captured using an idealized antenna setup. However, incorporating data preprocessing steps before inputting the data into a neural network can be beneficial. The advantages of these preprocessing steps will be explored in Chapter 3.

Parameter of Simulation Pipeline	
Parameter	**Value**
RADAR SETUP	
Operation Frequency	94GHz
Scatter Type	Optical Region
OBJECT DESIGN	
Object Classes	Cones, Cylinders, Spheres, Plates
Number of Classes	4
Number of Shapes per Class	32
Object Diameter Range	1cm - 10cm
RCS PARAMETER	
RCS Azimuth Sweep	[-180° : 0.1° : 180°]
RCS Elevation Sweep	[-90° : 0.1° : 90°]
Sampling Rate	0.1°
RADAR ECHO SIMULATION	
Number of Trajectories per Object Instance	1,000
Possible Rotation Rates (Azimuth)	[$\pm$0.05°, $\pm$0.1°, $\pm$0.5°, $\pm$1.0°]
Possible Rotation Rates (Elevation)	[$\pm$0.05°, $\pm$0.1°, $\pm$0.5°, $\pm$1.0°]
Initial Orientation	Random
Added Noise	
Fluctuation Model	Swerling Model 2
SNR of Additive Gaussian Noise	[5, 10, 15, 25]

Table 2.3: Parameters of simulation pipeline

3 Implementation and Comparison of Deep Learning Architectures

In this chapter, we are researching the application of various deep learning models for the classification of space debris objects, utilizing the radar dataset created in the previous section. Our primary objective is to examine which models perform best in interpreting monostatic radar time series data. These data samples capture the radar signal reflected by the target object, which returns back to the source of emission.

All deep learning models have been implemented using PyTorch, while the deep learning pipeline itself also utilizes PyTorch Lightning features [15, 16].

3.1 Possible Signal Processing Steps

The choice of signal processing steps before applying feature extraction can significantly impact the interpretation of radar signals. Especially for deep learning, different representations can have unique advantages and computational implications. The effect of data processing steps can be examined in Fig. 3.1, where representative data samples for each class are illustrated.

Voltage Over Time: This data format captures the raw time-domain signal received by the radar system and retains the original signal form. It directly represents the amplitude of the reflected radar signal as a function of time.

Power Over Time: Power representation involves squaring the amplitude (voltage) of the signal. This form is useful because the power of the signal is often more relevant than the voltage, especially in radar systems where signal strength (power) determines the ability to detect and classify objects.

Decibel (dB): Decibel is a logarithmic unit used to express the ratio of two values of power. In radar signal processing, converting signal power to dB can compress the dynamic range, making it easier to visualize and analyze changes. The conversion of power to decibels is typically performed using the formula:

$$dB = 10 \log_{10}(P + \epsilon) \tag{3.1}$$

where:

- P denotes the power level to be converted,

- ϵ is a small, positive constant introduced to ensure mathematical stability (to avoid the undefined logarithm of zero).

This logarithmic compression is particularly advantageous for architectures, such as ResNet and MLP, which often deal with radar signals exhibiting wide dynamic ranges. By adopting a dB scale, these models benefit from reduced sensitivity to extreme values, enhancing their numerical stability and overall performance.

Furthermore, logarithm scaling can enhance the contrast of low-amplitude signals, making subtle but potentially crucial differences more noticeable to the learning algorithm. Additionally, logarithmic scaling can lead to a more uniform distribution of error and performance across different signal strengths.

Normalized Decibel Conversion: Normalization standardizes feature scales, accelerating model convergence and enhancing learning efficiency. This process involves adjusting the data to a common scale by subtracting the mean and dividing it by the standard deviation. Standard deviation and mean are calculated based on the training set, leaving the validation and test set untouched to avoid potential biases.
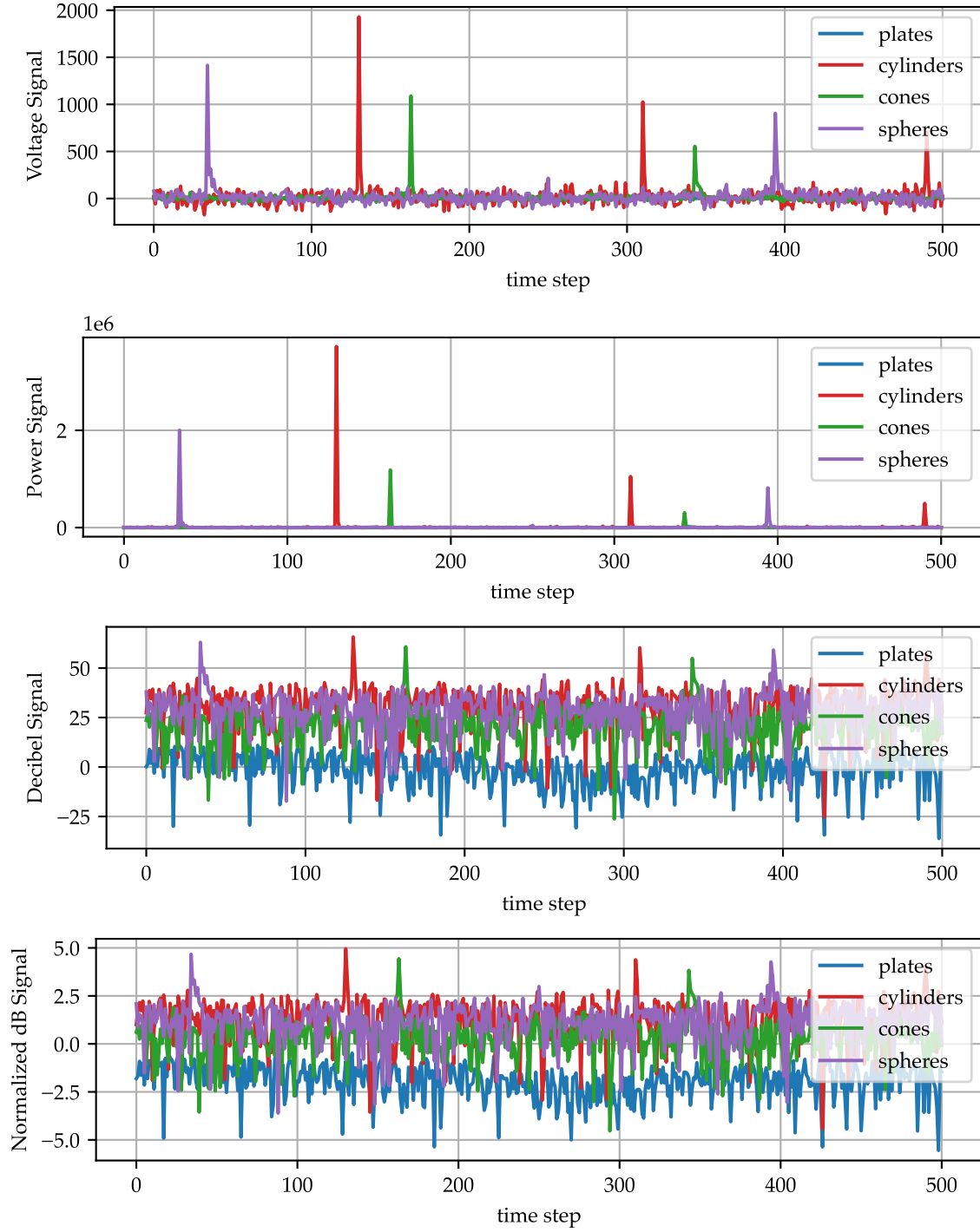
Figure 3.1: Signal processing steps on example sample instances

3.2 Deep Learning Pipeline

The deep learning pipeline begins by importing data from the simulation, converting them into tensors suitable for processing. An essential part of this procedure involves the elimination of undesirable zero samples.

Zero samples arise when the radar setup fails to detect objects or receive any radar return signals within the observation window. This issue often stems from an unfortunate angular position of the object in relation to the radar system, coupled with the object's shape and radar signature. Such circumstances can render the object undetectable during the observation period, leading to zero samples in the dataset.

This phenomenon, observed in datasets involving cones, resulted in up to 3,273 zero samples. The main issue is that these cones are poorly positioned towards the radar, and their pointed ends alter their radar detection signature.

To ensure the integrity and efficiency of the learning process, these zero samples were removed from the dataset. They contribute no meaningful information to the system.

The dataset was then divided into training, validation, and test sets following a 60-20-20 split ratio, employing a stratified split strategy to ensure an equal distribution of classes across all sets. This approach helps prevent biases and ensures a balanced learning process.

3.3 Comparing Different DL Approaches

Within the domain of machine learning, deep learning (DL) architectures are a common approach for performing classification tasks. However, their effectiveness varies significantly depending on the nature of the input data and the specific challenges presented by the task at hand.

This study explores various distinct models, including Multilayer Perceptron (MLP), Long Short-Term Memory (LSTM) [17], Residual Network (ResNet), and Transformer models equipped with either Sparse-Attention (SparseTRAN) [18] or Full-Attention mechanisms (FullTRAN) [19]. The MLP model serves as a baseline, offering a straightforward perspective on the complexity of space debris classification. We selected LSTM and Transformer models for their known capabilities in handling sequential data. The inclusion of the ResNet architecture is inspired by its successful application in related domains as highlighted in prior works [10].

The major objective of this experimental analysis is to identify the DL approach that maximizes accuracy in classifying space debris. In addition, the study examines the impact of various signal processing techniques on the performance of these models to optimize model outcomes.

3.3.1 Architecture Comparison and Impact of Signal Processing Approach

The accuracy of each model is a critical metric for evaluating the efficacy of different architectures. Considering that the class samples are evenly balanced across the training, validation,

and test sets through the use of a stratified split, no majority classes can negatively affect the accuracy measurements, allowing for a fair comparison of model performance.

To achieve optimal results for each model architecture, we engage in extensive experimentation to minimize multi-class cross-entropy loss during training. This process involves adjusting the hyperparameters unique to each architecture, as well as task-specific parameters such as the choice of optimizer, learning rate scheduler, number of epochs, and more. However, among these variables, the choice of the signal processing approach is just as important. Deciding to utilize the raw voltage signal or converting it to the decibel scale before normalization has a great impact on the performance of some models. This impact can influence multiple factors such as accuracy and generalization during training. The best performance outcomes from each model architecture are documented in Table 3.1.

Table 3.1: Best performing models of each architecture

Model	Accuracy	AvgPrec	Signal type	LR	Optim.	Scheduler	Epoch
LSTM	94.96%	98.57%	decibel	0.01	Adam	step	10
Bi-LSTM	94.83%	98.17%	raw	0.001	Adam	step	30
ResNet	94.76%	98.39%	decibel	0.001	Adam	step	10
Bi-LSTM	94.72%	98.46%	decibel	0.01	Adam	step	10
FullTRAN	93.93%	98.18%	raw	0.0001	Adam	plateau	50
SparseTRAN	93.33%	97.46%	decibel	0.0001	Adam	step	17
FullTRAN	89.84%	95.42%	decibel	0.0001	Adam	plateau	50
MLP	80.3%	86.87%	decibel	0.001	Adam	step	10

All models in Table 3.1 achieve accuracy levels above 80%, with the top performers nearing the 95% threshold. However, during experimentation and model tuning, we observed a significant influence of the signal processing technique on the effectiveness of these models.

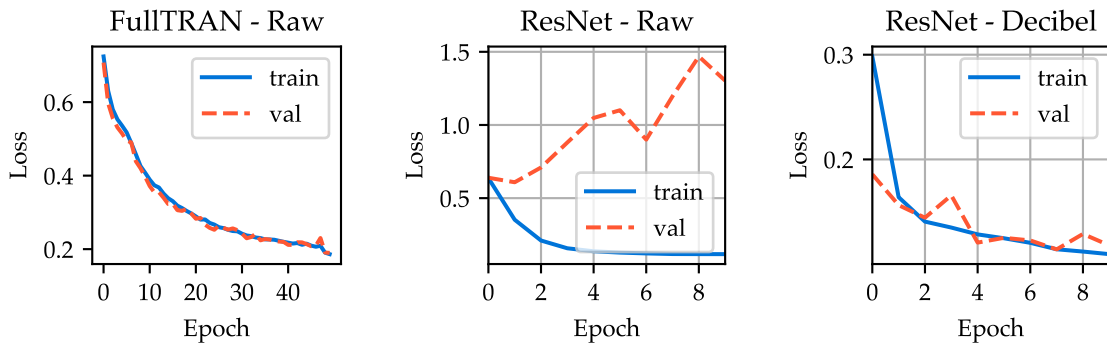**MLP and ResNet Generalisation Problem**

Figure 3.2: Loss curves of FullTRAN and ResNet vs. signal processing approach

FullTRAN demonstrates a strong resilience against the choice of data representation. Its performance metrics remain consistent regardless of the signal processing method employed. This includes FullTRAN's test accuracy, generalization abilities, and low training error. Unlike FullTRAN, MLP and ResNet encounter difficulties in generalization when applied directly to raw voltage signals, resulting in significantly lower accuracy on unseen data samples. During training, their validation loss diverges drastically. However, converting the raw signal to the decibel scale immensely enhances their generalization capabilities. This improvement is demonstrated in the case of ResNet, as shown in the loss curve comparison between the two signal processing choices in Fig. 3.2. During this comparison, we ensure the consistency of hyperparameter settings and use the same seed for training to guarantee a fair evaluation. The loss curve of FullTRAN is representative for both signal processing approaches since it performs equally well on both of them.

These performance differences observed in ResNet can be attributed to its capability to identify local correlations and capture hierarchical patterns (such as in images). This feature extraction can be hindered by outliers or significant voltage spikes, common in raw signal samples, potentially distorting the model's learning process by causing it to overfit to these anomalies rather than generalizing across broader patterns.

When the signal is converted to the decibel scale, it compresses the dynamic range, minimizing variance and decreasing the impact of outliers. Applying this logarithmic scale enables ResNet to leverage its convolutional layers more effectively without being distracted by outlier values, thereby improving its performance.

Adapting LSTM for Different Signal Processing Techniques

Long Short-Term Memory (LSTM) networks require different adjustments depending on the employed signal processing approach. Although LSTMs show effectiveness in handling both decibel-scaled and raw voltage signals, they encounter greater challenges in feature extraction from raw voltage signals. To achieve accuracy levels comparable to those obtained with decibel signals, several modifications are required. These include lowering the learning rate, augmenting the number of training epochs, and enhancing the model's complexity. These adjustments are essential to effectively narrow the gap in feature extraction proficiency between processing voltage and decibel signals, ensuring optimized performance across different signal types.

Table 3.2: Comparison of LSTM models

Name	Accuracy	Signal Type	Bidirectional	LR	Epochs
LSTM	94.96%	decibel	false	0.01	10
Bi-LSTM	94.83%	raw	true	0.001	30
Bi-LSTM	93.3%	decibel	true	0.01	10
LSTM	70.34%	raw	false	0.001	30

The outcomes of our experimental investigations, as detailed in Table 3.2, support our

claim, that LSTMs achieve quicker convergence and require less complexity to extract features from decibel signals. This efficiency is largely due to the decibel scale's ability to compress the dynamic range of signals, thereby enhancing the signal-to-noise ratio. This helps LSTM to focus on the temporal patterns rather than being distracted by extreme outliers. Furthermore, a more noise-reduced signal allows the LSTM's gates and cell states to learn and retain important features of the signal more efficiently.

Nonetheless, with specific adjustments, LSTM's performance on raw voltage signals can be significantly enhanced. By reducing the learning rate, the LSTM is prevented from stagnating at suboptimal optimization lows during training. Additionally, the incorporation of a more complex bidirectional LSTM architecture, which analyses data from both forward and reverse perspectives, enables the model to better capture patterns across all temporal steps. This approach improves the LSTM's ability to process and learn from raw voltage signals.

Slow Convergence of Full-Attention Mechanism

Experiments investigating Transformers which utilize Full-Attention mechanism (FullTRAN) have shown that these models are not influenced by the choice of signal processing. Unlike LSTM models, FullTRAN can extract equally well features from both decibel and raw signals without having to adjust any hyperparameters.

However, FullTRAN models require a significantly longer training duration to lower accuracy results in comparison to other deep learning models such as LSTM. To further explore these differences, we will examine the loss curves of both LSTM and FullTRAN over an extended training period. This detailed analysis is captured in Fig. 3.3, where we have increased the training epochs for LSTM applied on decibel signals from the initial 10, as used in our previous experiments (referenced in Table 3.1), to 30 epochs.
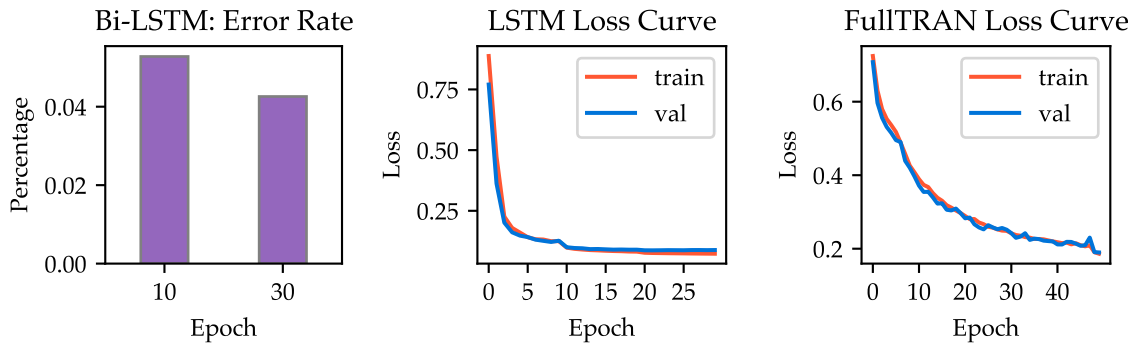

Figure 3.3: Convergence of LSTM (decibel) and FullTRAN (either)

In contrast to our observations with ResNet, where poor generalization was linked to reduced testing accuracy, Transformer models employing Full-Attention do not present the same issue. Instead, FullTRAN tends to converge more slowly, extending the required training time, and possibly reaching a performance plateau earlier than other models.

On the other hand, LSTM models converge more rapidly, especially when applied to decibel signals, achieving an accuracy plateau near 95% by the 10th epoch. This quicker convergence may be attributed to the smaller model size and the ability of LSTM to utilize a smaller learning rate in our classification task. Extending the training time of LSTM beyond 10 epochs merely leads to a negligible decrease of the error rate (1 - accuracy) by 1% as seen in Fig. 3.3.

Introduction of Sparse-Attention

The Full-Attention mechanism in the Transformer model not only suffers from slow convergence but is also afflicted by extensive training time. To address these issues, Sparse-Attention has been introduced to the world of deep learning as an efficient solution to mitigate these issues [18]. By selectively focusing on a subset of the input data rather than the entire dataset, Sparse-Attention significantly reduces the computational load and enhances memory efficiency. This focus on the most relevant data points also enables Sparse-Attention to accelerate model convergence.

Both FullTRAN and SparseTRAN undergo training with the same set of hyperparameters such as positional encoding, the dimensionality of embedding, and the number of layers, differing only in the Transformer encoder and the type of attention mechanism used.

The switch to Sparse-Attention results in amazing benefits: not only does it reduce the training time, but it also leads to an impressive accuracy of 93.3% by the 17th epoch. This outcome significantly outperforms the convergence speed of the Full-Attention variant, which fails to achieve similar accuracy within 50 epochs. Detailed metrics are available in Table 3.3.

The effectiveness of the sparse mechanism appears to be less robust against the signal processing choice in comparison to the FullTRAN model, as illustrated in Fig. 3.4. A possible explanation for this is that Sparse-Attention selectively focuses on specific parts of the data to reduce computational load. This focus might result in overlooking subtle yet critical features present in raw data formats. For instance, anomalies and outliers in raw voltage signals could negatively influence the attention mechanism, causing the model to neglect less obvious but vital patterns. Logarithmic scaling can reduce variance and diminish the impact of outliers, making subtle features more pronounced and easier for Sparse-Attention models to detect and learn from.

In contrast to Sparse-Attention, Full-Attention analyzes all aspects of the data with equal importance, avoiding any selective filtering. This approach demonstrates a comprehensive understanding of every detail within the data, including both outlier values and minor variations in the signal. When applied to raw voltage signals, Full-Attention demonstrates high accuracy during classification, showing no need to convert signals into a decibel scale, which has been a common requirement for many other models. This claim is supported by the experimental results highlighted in Table 3.3, revealing only a 3% difference in classification accuracy between Full-Attention applied on raw signals versus on decibel signals. This minor difference might be the result of logarithmic scaling, which smooths out certain features of the raw signal during the compression of the signal's dynamic range. Unlike other models that struggle with the raw signal's temporal or spatial structure, these inherent characteristics seem to improve FullTRAN's capability to accurately interpret signals.

Table 3.3: Comparison of Transformer models

Transformer Model	Attention	Signal type	Accuracy	Epochs	Runtime
SparseTRAN	Sparse	decibel	93.93%	17	**0h 25min**
FullTRAN	Full	raw	93.33%	50	**8h 7min**
FullTRAN	Full	decibel	89.84%	50	**8h 2min**
SparseTRAN	Sparse	raw	82.81%	17	**0h 22min**

Exploring the Effects of Changing Signal Processing Techniques

Our investigation dives into various deep learning architectures to assess the impact of the signal processing choice on their performance. Certain models, like LSTM, can be adapted to perform well on both decibel and raw voltage signals. However, some models exhibit a slight preference or are even restricted to a specific signal processing approach to maximize their classification accuracy.

To provide a comprehensive overview of which models favor which signal processing approach, we retrain the top-performing models as identified in Table 3.1 and apply alternative processing techniques to each of them. Throughout this training procedure, we maintain the same hyperparameters to guarantee a fair evaluation. The compiled results are depicted in Fig. 3.4, highlighting the influence of signal processing on a model's error rate.
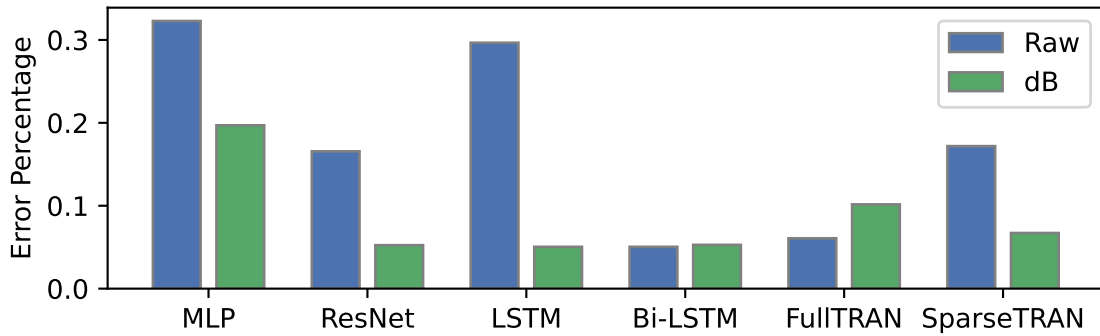
Figure 3.4: Comparing impact of signal type on error rate (1 - accuracy)

3.3.2 Model Implementation Parameter

MLP	
Hyperparameter	Value
Hidden Sizes	[256, 128]
BatchNorm	Yes
Activation	ReLU

LSTM	
Hyperparameter	Value
Hidden Size	128
Num Layers	2
Bidirectional	False/True

Transformer (SparseTRAN, FullTRAN)	
Hyperparameter	Value
Embedding Option	conv1d
Embedding Dim	256
Feedforward Dim	1024
Nhead	8
Number of Layers	8
Attention	Full/Sparse
Attn Window Size	100
Encoding	Positional

Figure 3.5: Model parameters of MLP, LSTM, and Transformer models

One Dimensional Residual Network

```
┌─────────────────────┐
│    1 x 3 conv        │
│  dim: 64, stride: 2  │
└─────────────────────┘
   [Max pool, stride 2]
┌─────────────────────┐
│ 1 x 3 Residual Block │  x 2
│ dim: 64, stride: 1, 1│
└─────────────────────┘
┌─────────────────────┐
│ 1 x 3 Residual Block │  x 2
│ dim: 128, stride: 2, 1│
└─────────────────────┘
┌─────────────────────┐
│ 1 x 3 Residual Block │  x 2
│ dim: 256, stride: 2, 1│
└─────────────────────┘
┌─────────────────────┐
│ 1 x 3 Residual Block │  x 2
│ dim: 512, stride: 2, 1│
└─────────────────────┘
     [Average Pool]
┌─────────────────────┐
│       fc 256         │
└─────────────────────┘
```
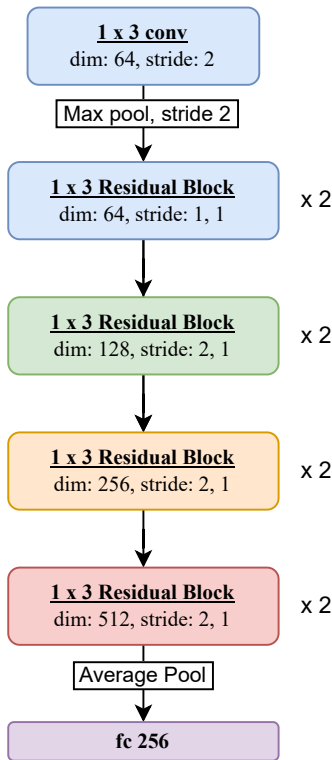
The Idea of implementing a 1D residual network was inspired by papers in related domains [10]. The implementation itself is flexible and allows specifying the number of blocks per layer, the activation function and more, providing a highly customizable framework.

Residual Block:
Each block consists of two convolutional layers with Batch Normalization and a SiLu activation, including a shortcut connection for residual learning.

Fully Connected Layer:
The fully connected layer transforms the pooled features into a feature vector, followed by Batch Normalization, a SiLu Activation, and Dropout (0.5) for regularization.

Figure 3.6: ResNet

3.3.3 Addressing the Misclassification Issue
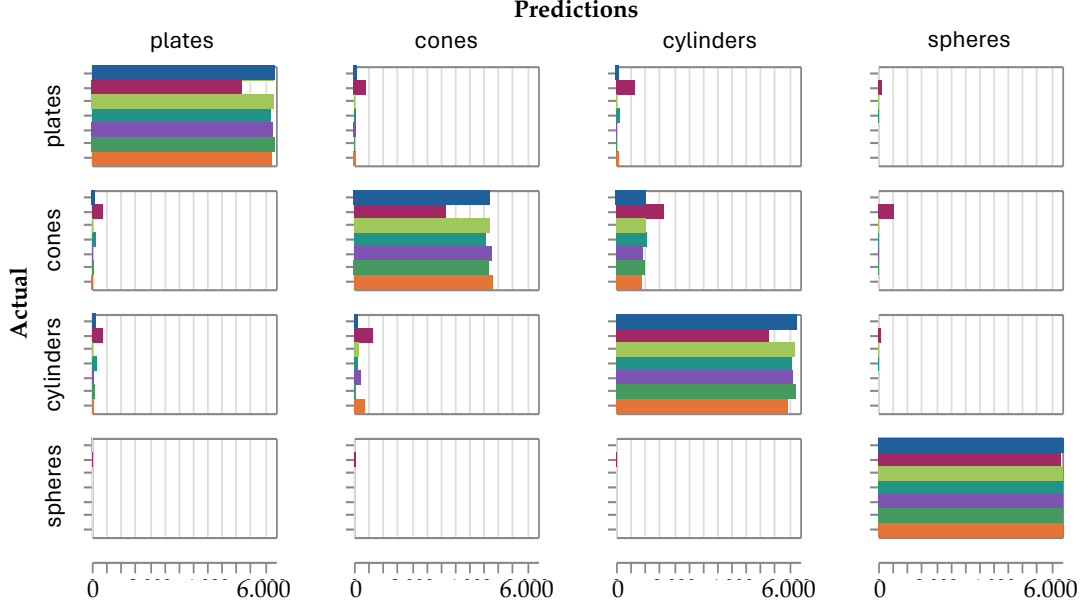
Figure 3.7: Confusion table representing models from Table 3.1

The misclassification of cones remains a persistent challenge across all examined deep learning architectures. It seems, that the same set of cones is continuously erroneously identified as circular cylinders. This issue raises a critical question: Is this anomaly rooted in the data or could the fault lie within the model's structure?

To investigate the root cause of this misclassification issue, it is essential to thoroughly analyze the simulated data samples and the performance of a representative deep learning model. Given that LSTM has demonstrated superior accuracy compared to other models, it will serve as the primary example to investigate the misclassification origin and to identify specific edge cases which lead to the erroneous classification of cones.

To better understand the quality of feature extraction and its impact on the model's performance, we need to examine the spatial distribution of the extracted features. These features are visualized in Fig. 3.8 on a two-dimensional plane through Uniform Manifold Approximation and Projection (UMAP) [20]. This technique is particularly useful for projecting high-dimensional data onto a two-dimensional plane, offering a clearer insight into the feature space.
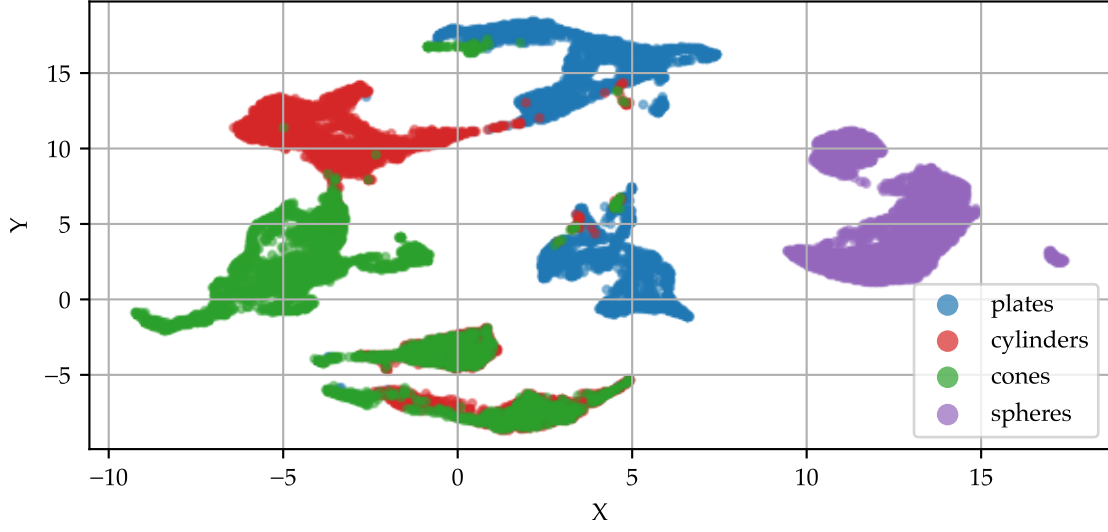
Figure 3.8: Feature map of LSTM (UMAP)

In the feature map visualization, instances of cones and cylinders are mostly separated into two distinct clusters, demonstrating clear differentiation for the majority of cases. However, there seems to be an additional cluster at the bottom of the plot, where specific cylinders and cones appear to converge within the same feature space. This observation indicates that, under certain conditions, some data samples of the cone and cylinder class are too similar to be easily distinguished, likely leading to the misclassification issue observed.

(a) Clear separation   (b) Problem area
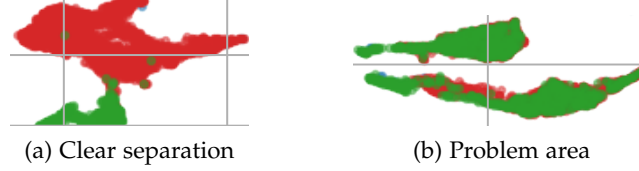
Figure 3.9: Cones and cylinders in feature space

To investigate this claim and to understand, why certain samples of cones and cylinders are mapped to the same feature space, we need to examine the original data input of the LSTM model. Therefore, we are plotting two comparisons visualized in Fig. 3.10. The first plot showcases a pair of instances (one cone and one cylinder) that are closely aligned in the feature space within the problem area. The second visualization contrasts a pair of instances that are maximally separated within feature space, each correctly positioned within their respective clusters. In other words, we are comparing a similar pair and a dissimilar pair of cone and cylinder instances.

The side-by-side comparison of these plots strongly suggests that the problem of misclassification arises from the data samples themselves, instead of a faulty feature extraction by the LSTM model.

(a) Similar samples from Problem Area

(b) Max. separated in feature space
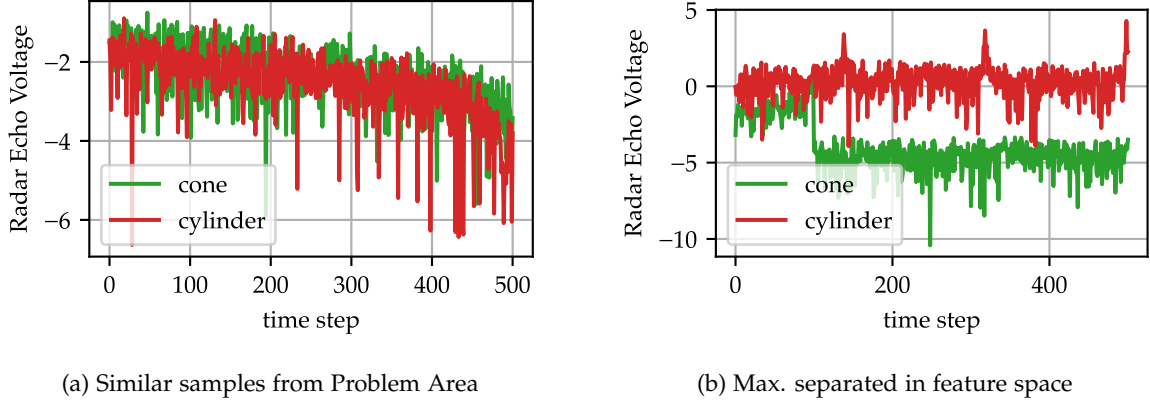
Figure 3.10: Decibel signals of similar and a dissimilar cone and cylinder set

However, a key question remains: Under which circumstances do the instances of cones and cylinders become indistinguishable?

The answer lies in the radar cross-section (RCS) Signature of each target object and the limited observation time of 501 time steps, in which we can detect the radar echo reflected by the target instance.

The RCS signature essentially describes an object's ability to reflect radar signals back to the source, which is heavily dependent on its shape, especially when the radar operates in the optical region. Therefore, we need to explore dimension-specific shape parameters of cones and cylinders and investigate which parameters lead to similar RCS Signatures for the two classes. Focusing on the parameter set of radius and height, our analysis identifies two interesting shape profiles: These two profiles are labeled as "wide" ($r \geq h$) and "tall" ($r \ll h$) are illustrated in Fig. 3.11.

(a) *Radius $\geq$ Height*
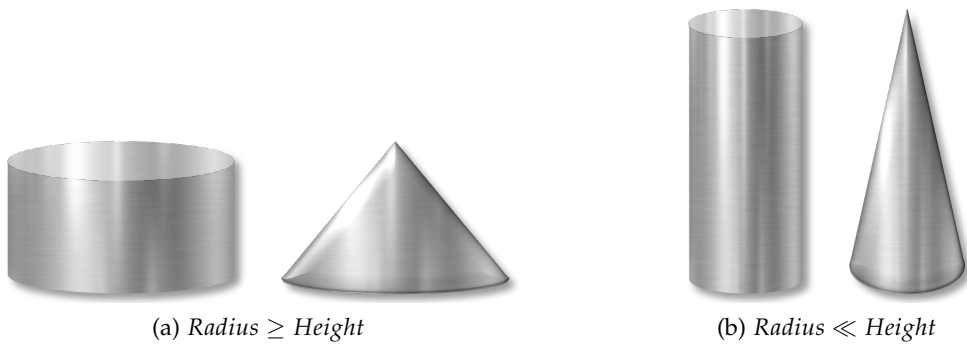
(b) *Radius $\ll$ Height*

Figure 3.11: Cones and cylinder: wide vs. tall

The parameters of height and radius significantly influence the shape of these objects, but the question remains: which specific shapes result in similar RCS signatures for cones and cylinders? Two key observations emerge from Fig. 3.11.

Initially, one might expect that the similarity between cones and cylinders stems from their shared circular base. However, this alone does not account for why misclassifications mostly involve cones and not circular plates as well, which also share this feature. This discrepancy indicates that the root cause of the misclassification extends beyond just the circular structure.

The second observation sheds more light on the issue: as cones become taller, the angles of their sides approach parallelism. This structural change makes the lower flat portion of an elongated cone appear similar to that of a cylinder.

(a) Sharp corner   (b) Blunt corner
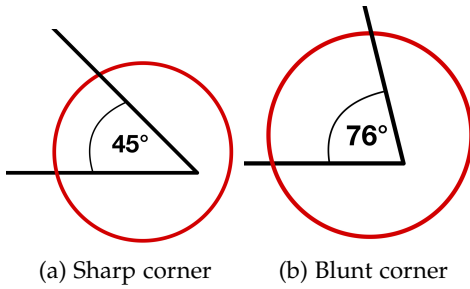
Figure 3.12: Sharp vs blunt corner

As previously discussed, the RCS signature, which is critical for classification, is primarily influenced by the shape of an object in the optical region. This means that RCS signatures are mostly distinguished by the presence of edges and corners on an object's surface. This is interesting, because while elongating cones, their sides not only become closer to parallel, but this also results in the base angle at the circular base approaching 90 degrees, much like that of cylinders. This phenomenon is visualized in Fig. 3.12, where we differentiate between "sharp" cones (radius >= height) and "blunt" cones (radius « height). By focusing our attention on these corners, we seem to have found first evidence to support the claim, that tall cones under specific circumstances are misclassified as cylinders.

To find further evidence to support this hypothesis, it is necessary to analyze the RCS signatures of both tall and wide cones and then compare these to the RCS signature of a cylinder sharing the same height and radius parameters. The following illustrations display these RCS signatures across an azimuth spectrum ranging from -180° to 180° and an elevation spectrum from -90° to 90°. At first, we will analyze the RCS signature of wide cones in Fig. 3.13, which we assume to appear differently from the signature of cylinders.

(a) RCS cone   (b) RCS 3D cone   (c) RCS cylinder   (d) RCS 3D cylinder
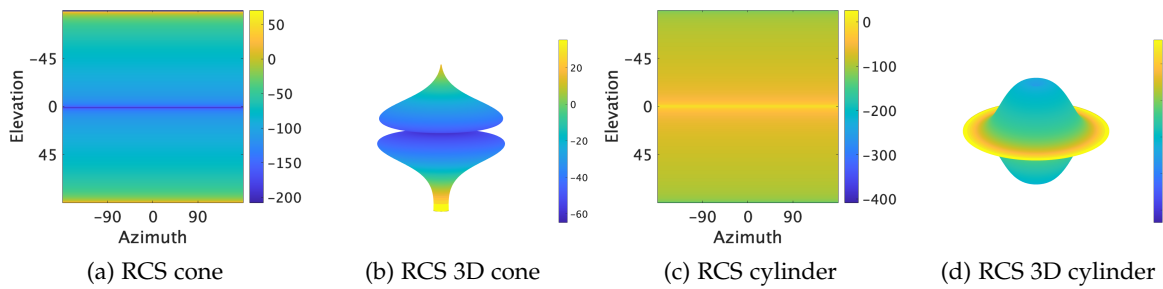
Figure 3.13: RCS of wide cones and cylinders

Given the complete RCS Signature of both classes, distinguishing between wide cones and cylinders seems to be straightforward, supporting our claim that the circular structure alone is not the primary factor behind the misclassification issue. In the following Fig. 3.14, we will examine the more interesting case of tall cones which have blunt base corners.

(a) RCS cone   (b) RCS 3D cone   (c) RCS cylinder   (d) RCS 3D cylinder
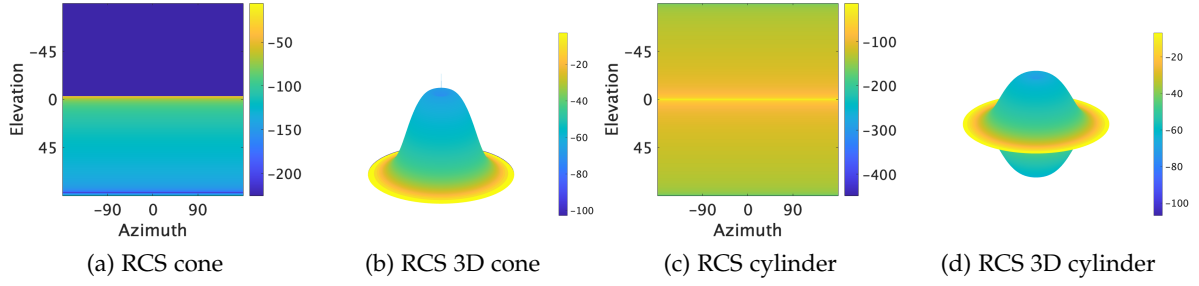
Figure 3.14: RCS of tall cones and cylinders

As anticipated, the shape's edges significantly influence its RCS signature. As the base corners of a cone become blunter, a portion of its RCS signature begins to resemble that of cylinders, specifically within the elevation spectrum of [0°, 90°]. This range extends from the horizontal side perspective of a cone at a 0° elevation angle to the top perspective at the circular base's center at a 90° elevation. This spectrum includes the blunt corners, which we assume to be the main driver behind the misclassification issue.

However, this similarity in the RCS signature within the elevation spectrum of [0°, 90°] alone cannot explain the misclassification of some cones. The distinct differences within the elevation spectrum of [-90°, 0°] suggest that additional factors play a crucial role such as the observation window of the radar system and the target object's motion profile.

The reflected signal captured by our radar setup is one-dimensional and constrained by an observation window of 501 time steps. The object's relative positioning and rotation pattern during motion determine which parts of the RCS signature are detected. Under ideal conditions, the complete RCS signature of an object can be observed. However, in less favorable situations, we may only catch a partial glimpse, affecting classification accuracy.
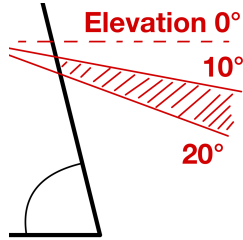
Figure 3.15: Elevation spectrum [10°, 20°]

To explore one such challenging scenario, we need to recreate a situation where the motion profile and observation window constraints limit our view to a section of the RCS signature which can either belong to a cylinder or a cone. In this scenario, we maintain a constant azimuth spectrum of [-180°, 180°] but apply a restriction on the elevation spectrum to a narrow interval of [10°, 20°]. This interval is selected from the previously identified potential problem area for accurate classification. The illustration of this constrained scenario is provided in Fig. 3.16.

We have identified clear evidence showing under which conditions the radar signatures and the resulting reflected radar signals of cones and cylinders become indistinguishable. This evidence, presented in Fig. 3.16, supports our assertion that the misclassification of cones originates from the specific RCS signatures of tall cones coupled with unfortunate rotational
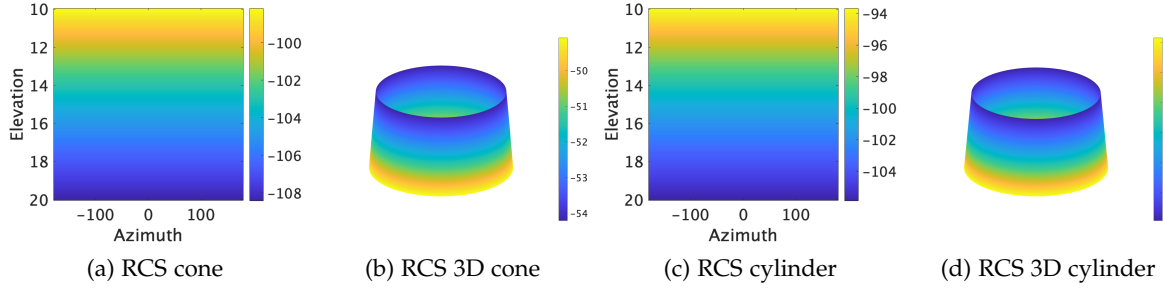
<div align="center">

(a) RCS cone   (b) RCS 3D cone   (c) RCS cylinder   (d) RCS 3D cylinder

Figure 3.16: Tall cones and cylinders with elevation spectrum [10°, 20°]

</div>

patterns. These patterns restrict observations to a narrow elevation range between 0° to 90°, leading to misclassification.

To translate these findings into feature space, we need to leverage the previously employed pretrained LSTM model for feature extraction. The process is illustrated in Fig. 3.17, which features additional simulated samples representing similar scenarios to the one depicted in Fig. 3.16. These samples specifically highlight elongated cones at a consistent elevation of 10°, thereby narrowing our observational scope of the target object.

As expected, these samples are mapped into the previously identified problematic area within feature space, confirming our hypothesis once more.

<div align="center">
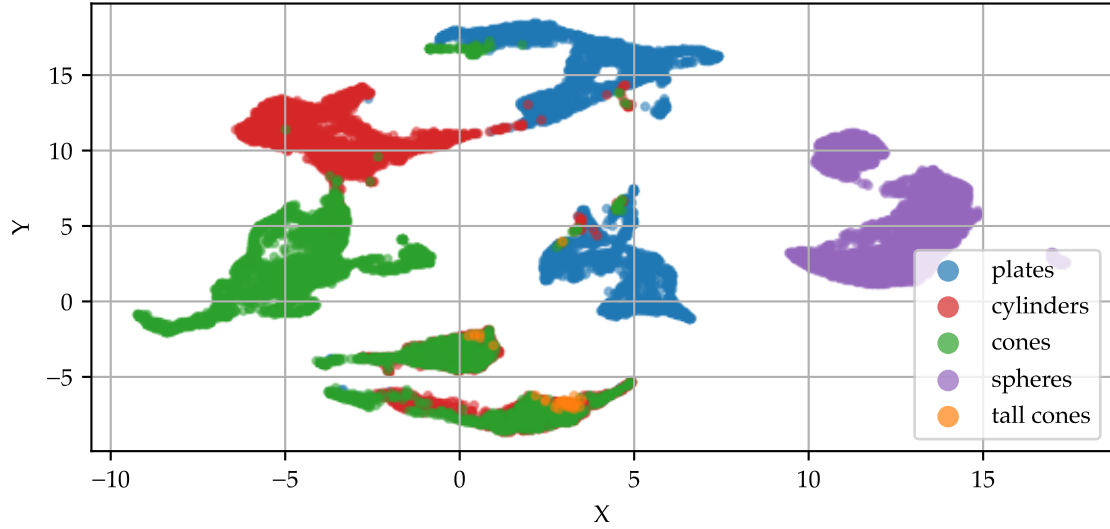
Figure 3.17: Feature map of LSTM (UMAP) including tall cones

</div>

Since the misclassification of cones arises from factors beyond the control of our deep learning model, this is a difficult problem to address by solely using machine learning tools.

Improvements in accuracy are more likely to result from modifications to the radar setup rather than adjustments within the machine learning model. In certain scenarios, simply

expanding the observation window could enhance detection accuracy. However, this solution might not be universally effective for all tall cones, as the elevation angle of the target's orientation might remain constant or vary only within the problematic range of 0° to 90° elevation. A potentially more effective approach involves the use of multistatic radar systems. By deploying multiple radar sensors and antennas, these systems introduce additional entropy through varied perspectives on the target object. This multiplicity of viewpoints is more likely to significantly enhance the overall accuracy of the radar system.

4 Evaluating Robustness against Radar Distortions

The original training data generated by our simulation pipeline already incorporates fluctuation in the radar signature of each object due to vibration and additive Gaussian noise to approximate various noise types. However, capturing radar signals in real-world applications can introduce additional radar-specific distortions, such as signal occlusion, clutter, sensor saturation, subsampling, or a combination of these factors.

Given the expensive computational cost of simulation, we are evaluating the robustness of our pretrained model exclusively on unseen data by modifying the existing test set. In this robustness evaluation, we are assessing accuracy relative to the intensity of our radar noise factor by examining each radar-specific distortion type individually. By introducing these artificial distortions, we aim to assess the resilience of deep learning models to real-world data imperfections.

However, careful consideration is necessary when evaluating robustness against radar distortions! Many of our examined deep learning models are designed to operate on a decibel scale, whereas the original captured signal is in the form of voltage. Given that all radar distortions occur at the raw signal level, it's crucial to perform any modifications on this raw voltage signal. Only after these adjustments should the data be converted to a decibel scale if the specific architecture being evaluated requires it.

To ensure the reliability of our experiments, each robustness test is conducted five times. This procedure is designed to mitigate the impact of any irregular or random behavior exhibited by the deep learning models. Averaging the results can provide a general sense of model performance across tests, while the median can be more resilient to outliers, offering a more stable measure of central tendency. Given that the following experiments show only slight variations in their outcomes, using the average will more accurately reflect the consistent performance of the models across different test iterations.

4.1 Occlusion

The core idea behind random occlusion is to inject a degree of unpredictability into the dataset, reflecting the irregular and often unforeseeable nature of capturing radar signals in real-world scenarios.

Occlusion can take various forms. For instance, setting time steps to zero represents a binary form of occlusion, indicating either the presence or absence of data. It can also be more nuanced, with signals being partially obscured or degraded, rather than completely

missing. However, such situations demand a deeper comprehension of the specific conditions at play.

In space environments, sensors may lose track of space debris due to unforeseen factors such as shadowing, technical failures, or interference. To address these unforeseen challenges, our study focuses on two types of binary occlusion: Random Dropouts and Block-wise Dropout.

4.1.1 Random Dropouts

Random dropouts simulate the occlusion of data points within time series data by setting random time steps to 0, similar to packet loss during data transmissions. The advantage of employing a random pattern lies in its ability to approximate a wide range of occlusion scenarios. At this point, it is important to remember, that we add an offset of 1e-10 to the signal prior to signal processing, to ensure mathematical stability during the logarithmic decibel transformation (see Eq. (3.1)).

During this evaluation, the intensity of these random dropouts is scaled by adjusting the percentage of missing data points, enabling the evaluation of model robustness across a spectrum of occlusion severities. This process provides valuable insights into how models trained on raw voltage signals are more resilient against this kind of distortion, maintaining better accuracy results than their decibel trained counterparts.
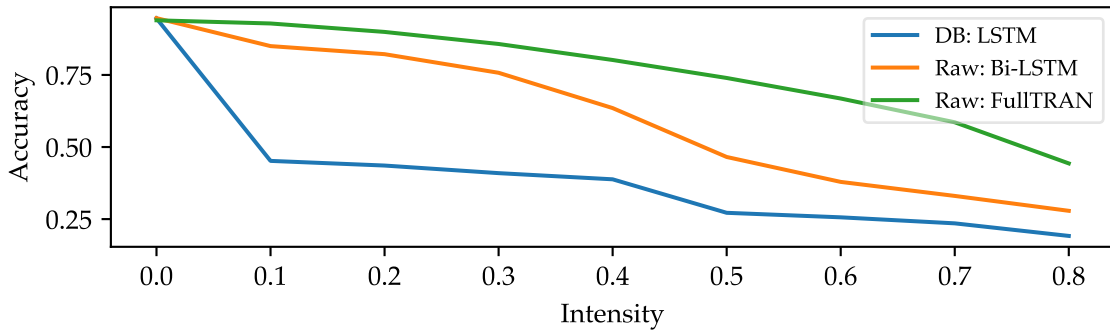
Figure 4.1: Accuracy distorted by Random Dropout

In Fig. 4.1, LSTM represents the overall bad performance of all deep learning architectures trained on decibel signals, illustrating the damaging effect of binary occlusion on their classification abilities. When applying an occlusion intensity of only 10%, these models suffer a dramatic decrease in accuracy of approximately 50%. However, this decline in performance is not primarily due to a loss of information during binary occlusion. Models trained on raw voltage signals are less affected by this distortion. The disparity in resilience between the two approaches must therefore be linked to how these models handle the signal's dynamic range and how this affects the applied feature extraction.

Fig. 4.2 further illustrates the impact of binary occlusion on raw and decibel scaled signals. Recognizing the effect of binary occlusion on raw signals is straightforward, while interpreting
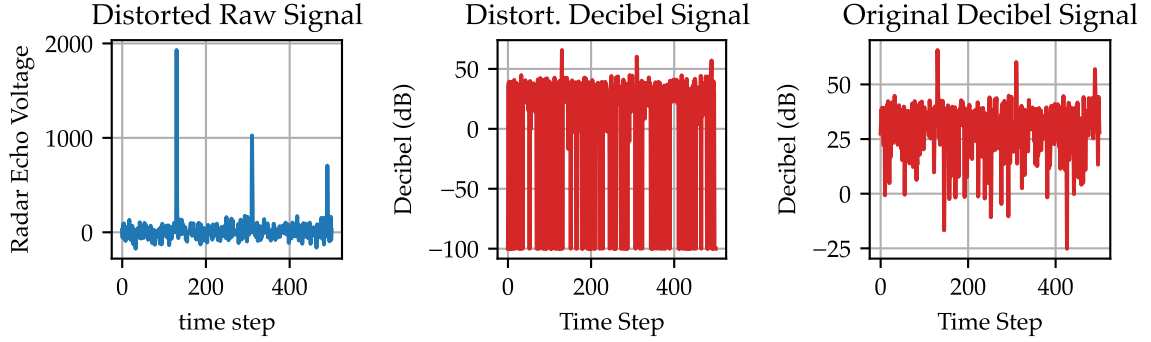
Figure 4.2: 10% Random Dropout applied on cylinder signal

occlusion after the complex alterations performed by the logarithmic transformation appears to be more complex. This is especially true for Block-wise Dropouts, illustrated in the following section by Fig. 4.4.

Therefore, models trained with raw signals, such as Bi-LSTM and FullTRAN, benefit from the signal's full dynamic range and original feature representation, making them more robust against binary occlusion. In contrast, models trained on the decibel scale experience way higher levels of distortion on a logarithmic scale, even if the actual occlusion in the radar environment was not as intense.

4.1.2 Block-wise Dropout

Block-wise Dropout refers to a specific type of occlusion, in which radar distortions zero out contiguous blocks of time steps. This evaluation focuses on the classifier's ability to maintain its performance in the face of unforeseen information loss, similar to the scenarios encountering Random Dropouts.
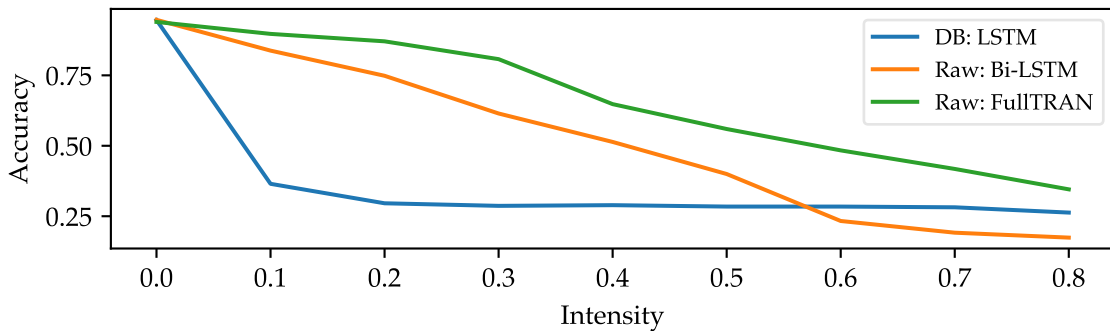
Figure 4.3: Block-wise Dropout and its effect on classification performance

The intensity of occlusion is determined by various occluded percentages of the total signal

length, represented by consecutive missing data points in each test data sample. The locations of these dropped intervals are selected at random. This process supports the previous claims of the "Random Dropout" Section, illustrating how decibel trained deep learning models are more affected by occlusion, compared to those trained on raw voltage signals.
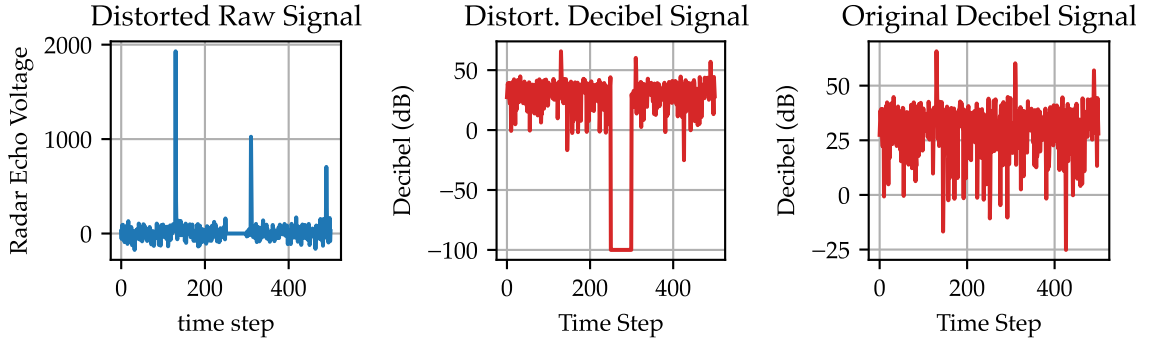
Figure 4.4: 10% Block-wise Dropout applied on cylinder signal

The Fig. 4.3 supports our previous claim that Block-wise Dropout appears to have a similar impact on model performance as the previously tested "Random Dropouts" experiment. Models trained on raw voltage signals signal tend to directly identify occlusion windows in their data representation. In contrast, models trained on decibel signals struggle to interpret the originally simple alterations caused by occlusion after a signal undergoes logarithmic transformation.

However, both model groups exhibit worse occlusion resilience when occlusion is applied as continuous windows rather than random data points. Although the same number of time steps are omitted, the loss of continuous windows makes pattern recognition more difficult. Therefore Block-wise Dropout is associated with a greater loss of entropy for the system, which explains the strong impact on the classification task.

4.2 Sensor Saturation

Sensor saturation occurs when a sensor receives a signal that surpasses its maximum recording capability, which can arise from overly strong signal sources or improper sensor gain settings.

When saturation occurs, the recorded signal is clipped or flattened at the saturation points, leading to a distortion of the signal's true amplitude or shape during those intervals.

To simulate artificial saturation, there are two interesting methods. **Percentile-Based Thresholds** are closely aligned to the actual data distribution. Setting thresholds at high percentiles, ensures that only the most extreme values are impacted, mirroring real-world saturation events. On the other hand, **Absolute Value Thresholds** require precise domain knowledge to identify specific signal levels at which saturation commonly happens. Given the unpredictable nature of radar applications in space, opting for Percentile-Based Thresholds

offers a more realistic simulation of saturation by adhering to the natural distribution of signals.

The robustness evaluation is performed by modifying the test set using Percentile-Based Thresholds across various intensity levels. For instance, a 20% intensity implies that 20% of the data points within a sample are cut off to the predetermined saturation threshold, reflecting how saturation could realistically affect data. In contrast to our observations during the occlusion experiments, decibel signals appear to be the better feature representation when dealing with sensor saturation. While this difference in model performance is not as extreme as during occlusion, the impact of the selected signal processing approach is still visible in Fig. 4.5.

In this illustration, we are going to compare the results of the saturation experiment based on three models: LSTM and SparseTRAN both trained on decibel signals, and Bi-LSTM trained on raw voltage signals.
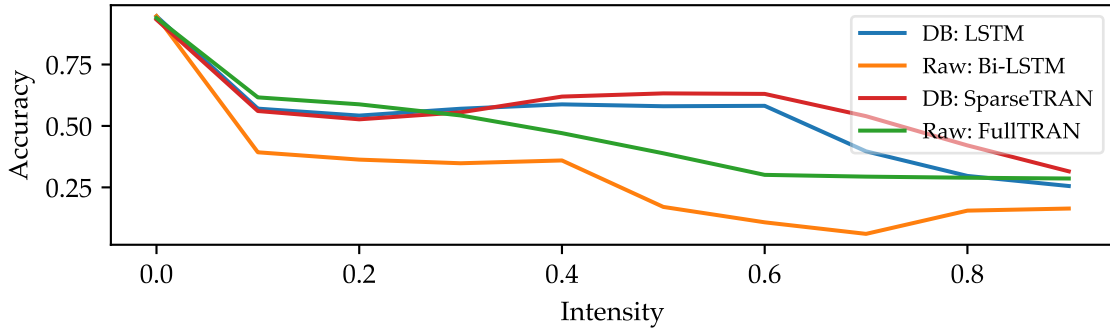
Figure 4.5: Percentile Sensor Saturation examined via accuracy

Interesting observations include, that despite LSTM (decibel) and Bi-LSTM (raw) showing equivalent performance during training, LSTM utilizing the decibel scale appears to be more resilient against saturation up to a specific threshold. This pattern of resilience is similarly observed when comparing the two Transformer models, SparseTRAN (decibel) and FullTRAN (raw), indicating the potential benefits of training models on decibel-transformed data. To find evidence for this claim, we will have to examine the effect of saturation on our signal data.

In raw voltage signals, saturation is characterized by a linear clipping process, as illustrated in Fig. 4.6. However, transitioning to a decibel (dB) scale can help in distinguishing features within saturated segments that would otherwise appear flat and undifferentiated in a linear scale.

Furthermore, we can observe varying degrees of performance across the spectrum of decibel-trained deep learning architectures. Especially models using Sparse-Attention mechanisms such as SparseTRAN appear to be more resilient against sensor saturation, even though they have not been trained on saturated data. Following a significant decline in accuracy by 30%, SparseTRAN not only demonstrates stabilization but also appears to regain 10% of
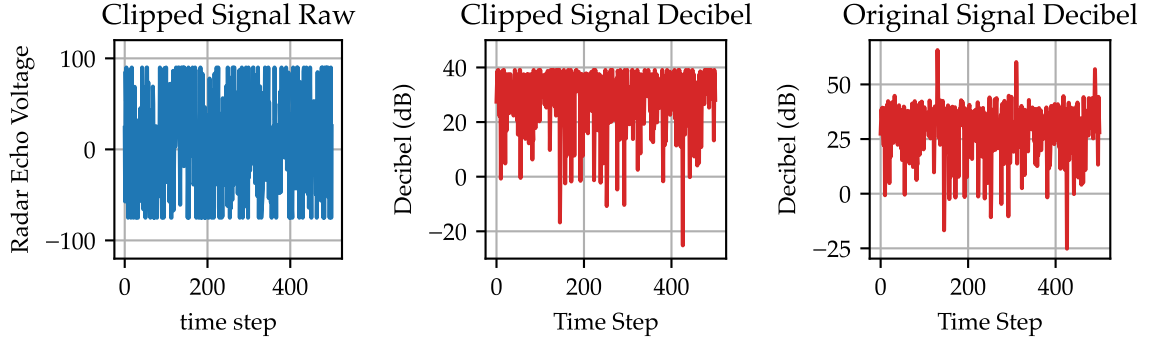
Figure 4.6: Sensor Saturation applied on cylinder signal

its accuracy, even as the intensity of the saturation distortions increases. This trend persists until reaching a threshold of 60% saturation. As sensor saturation intensity escalates, the Sparse-Attention mechanism appears to adapt and mitigate some of its effects. It probably achieves this by prioritizing the analysis of unsaturated segments over the distorted ones, thereby enabling a partial recovery of accuracy.

4.3 Clutter

In radar systems, clutter refers to unwanted echoes or reflections from objects other than the target of interest, such as terrain, buildings, or atmospheric phenomena. In the context of space, particularly during the classification of space debris in Low Earth Orbit (LEO), clutter can arise from multiple sources. Among these are other radar systems emitting electromagnetic signals, which can interfere with our observations. Additionally, the presence of fragmented debris creates a situation where our radar signals are reflected off multiple objects, adding complexity to the classification task. This environment highlights the significant challenge of precisely identifying individual pieces of debris against a diverse array of reflective sources.

These clutter signals add noise patterns to the captured signal, making it harder to isolate and identify the target signal. To evaluate robustness against clutter, we introduce both random and structured noise to simulate clutter scenarios. This method tests the model's capability to disregard these distractions and concentrate on the target signal. In the upcoming experiment, we will employ "Random Peaks" and sinusoidal signal patterns ("Sinusoidal Clutter Pattern") at various frequencies. This approach aims to understand their impact on our accuracy in the classification task, providing insights into the system's effectiveness in real-world clutter conditions.

4.3.1 Random Peaks

In this experiment, our goal is to evaluate the model's resilience to a rising number of outliers and anomalies, which clutter our target signal. We are investigating various levels of clutter intensity, where intensity denotes the number of time steps impacted by clutter outlier values. For each sample, the magnitude of the peaks is individually assessed and follows a uniform distribution. These amplitudes range from twice to four times the maximum amplitude of the raw signal sample, measured in voltage over time of the target signal.
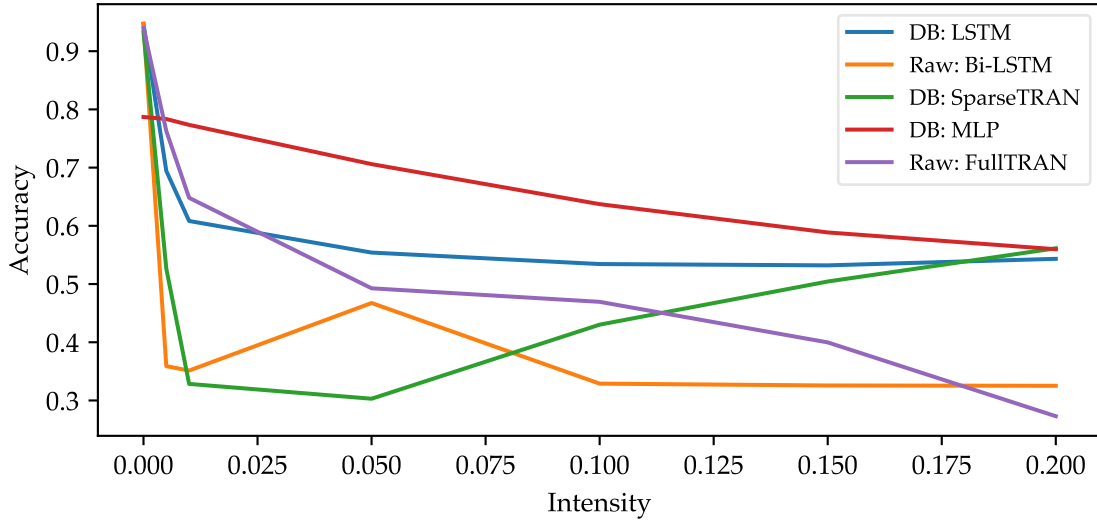
Figure 4.7: Random Peaks and its effect on classification performance

In the various situations we have analyzed, that transformations to the decibel scale have showcased denoising capabilities, enhancing the signal-to-noise ratio. This trend continues in our anomaly detection experiment. As illustrated in Fig. 4.7, models that employ signal processing to achieve a logarithmic scale exhibit greater resilience against increasing occurrences of anomalies. Especially MLP trained on decibel signals displays significant resistance to these distortions, even though it is known for its low complexity. This resilience can be attributed to the global perspective of MLP combined with decibel scaling, where decibel transformation compresses the dynamic range of the signal, enabling easier identification and disregard of anomalies.

A particularly interesting case is presented by the Sparse Transformer (SparseTRAN). Its Sparse-Attention mechanism demonstrates enhanced robustness to clutter as the number of anomalies increases in our experimental setup. Typically, our simulated radar signals include only 2-5 outlier values, based on the current position of the target object. If we only add 4 additional artificial outliers, it is difficult to assess if the target signal has been inflicted by noise and therefore to distinguish between inherent outliers and anomalies. However, the scenario shifts when 20 anomalies are introduced within an attention window of the Sparse-Attention mechanism. Under these conditions, the model is likely to deduce a
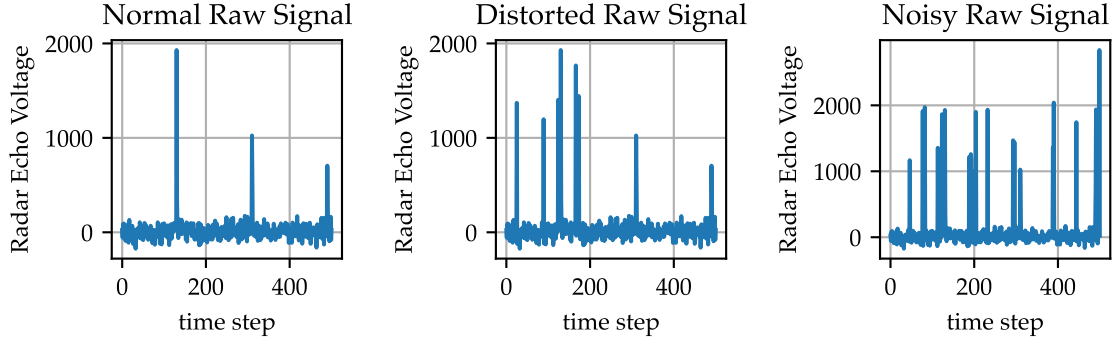
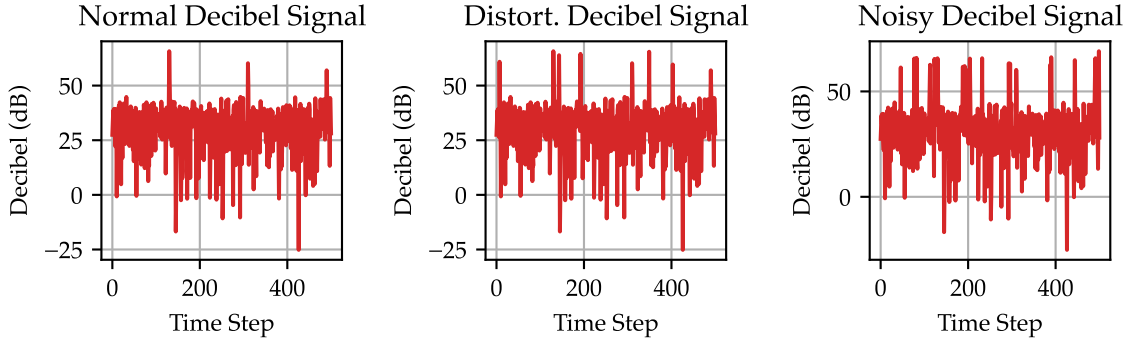Figure 4.8: Increasing Random Peak applied on raw cylinder signal

Figure 4.9: Effect of increasing Random Peak on decibel scale of cylinder signal

lower signal-to-noise ratio, prompting it to identify these peaks as noise and consequently shift its focus to more pertinent features. The application of a logarithmic scale further highlights these differences between standard signal components and anomalies, enabling the Sparse-Attention mechanism to more effectively prioritize critical signal features.

4.3.2 Sinusoidal Clutter Pattern

The sinusoidal clutter pattern can be modified in terms of the two parameters frequency and signal amplitude. The signal amplitude is adjusted for varying signal-to-noise ratios (SNR) of each individual signal. Frequencies are selected for their relevance to the radar signal's operational frequency, specifically targeting frequencies close to and multiples of the radar's 95 GHz operating frequency within the millimeter-wave range. This method is designed to replicate realistic interference scenarios which a radar system operating at such high frequencies may face, including challenges like harmonic interference, adjacent channel interference, and various other potential noise sources. Operating in the high frequency

spectrum, the system is characterized by its high-resolution and short-range capabilities. This makes it sensitive to interference and clutter in the gigahertz range and is less affected in the megahertz range or far beyond its operating bands.

Table 4.1: Radar Test Frequencies

Test Case	Frequency (Hz)
5% above radar frequency	9.87×10^{10}
1 GHz below radar frequency	9.3×10^{10}
Half of radar frequency	4.7×10^{10}
1 GHz (other sources)	1.0×10^{9}
Double the radar frequency	1.88×10^{11}

The key observation from studying the impact of sinusoidal cluster patterns on model performance is that models employing the decibel scale as a signal processing technique tend to exhibit less degradation in performance at low SNR levels. This is primarily attributed to the signal processing's ability to filter out noise when converting signals to a logarithmic scale. Contrary to these findings, models that apply a Full-Attention mechanism outperform all other deep learning models in terms of robustness. This superior performance is especially interesting since these models are trained on raw signals.
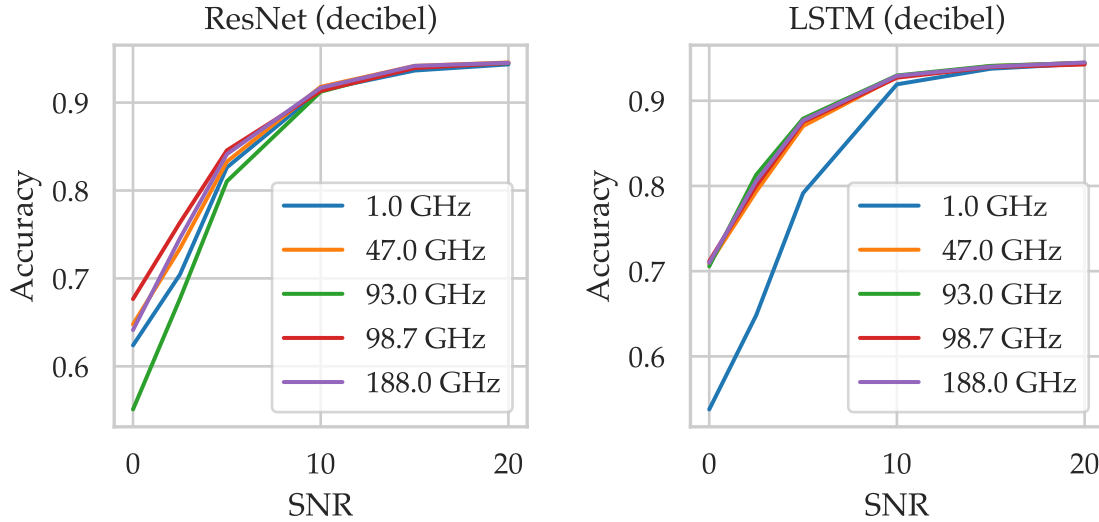
Figure 4.10: Models trained on decibel signal distorted by sinusoidal patterns

Models trained on decibel signals demonstrate efficiency in isolating the target signal, achieving near-complete separation with a signal-to-noise ratio (SNR) of approximately 10. This effectiveness is illustrated in Fig. 4.10. However, these models significantly differ in their sensitivity to the chosen noise frequency.

For instance, ResNet models exhibit a unique sensitivity to different frequencies when the SNR is lower than 10. This variance in performance might be linked to the inherent nature of ResNet's convolutional layers, which act as signal processing filters during the feature extraction process. This mechanism, designed to capture spatial-temporal features in the data, may respond differently to noise frequencies due to their localized nature. Consequently, certain frequencies of noise might align with or interfere with the pattern recognition capabilities of these local kernels, directly impacting the model's ability to accurately classify underlying radar signals.

However, LSTM demonstrates a unique resilience to noise across various frequencies, with their performance remaining relatively consistent whether the noise frequency is around the model's operating frequency or significantly higher. We link this uniform response largely to the gate mechanism intrinsic to LSTM units. LSTMs are designed to selectively remember patterns over long periods, thanks to their complex architecture including forget gates, input gates, and output gates. This gating mechanism ensures that irrelevant information (which could include certain frequencies of noise) is filtered out, while important features are retained and passed through the network. However, the exception to this rule arises with the noise frequency of 1 GHz, much lower than our operating frequency, which drastically decreases the model's performance. Low-frequency noise represents a slow-changing signal component, leading to confusion within the model's gate mechanism, which is supposed to filter out irrelevant short-term fluctuations.
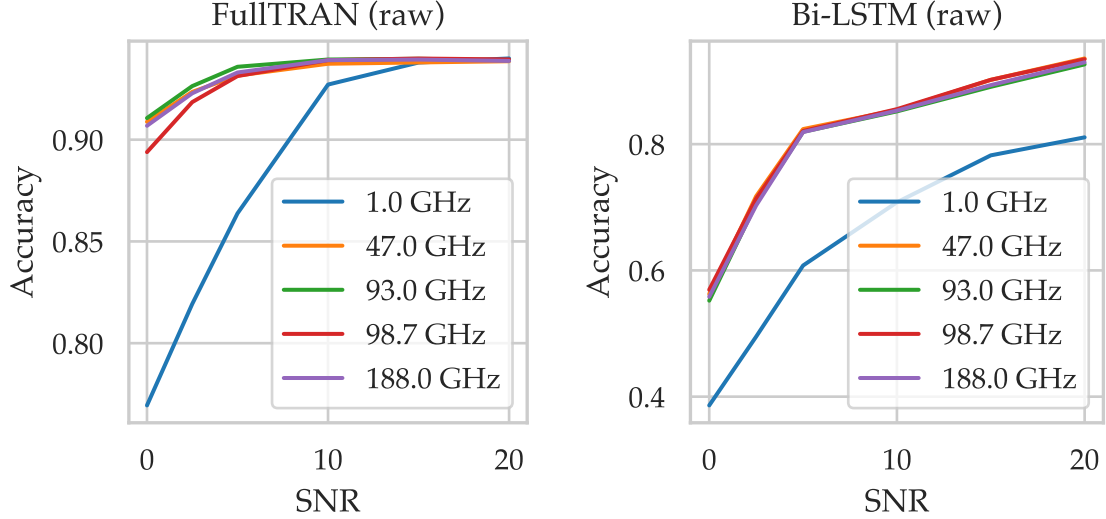

Figure 4.11: Models trained on raw voltage signal distorted by sinusoidal patterns

Transformer models like FullTRAN, employing a Full-Attention mechanism, showcase exceptional robustness against various noise frequencies, maintaining classification accuracy above 90% even with SNRs below five. This model type excels at filtering high-frequency

noise due to its ability to globally assess and weigh all parts of the input data, effectively distinguishing relevant signals from noise. However, it is less effective against low-frequency noise, such as at 1GHz, which is significantly lower than our operating frequency. This limitation might arise because low-frequency noise can blend with genuine signal patterns, making it harder for the attention mechanism to differentiate between noise and meaningful information. High-frequency noise, with its rapid changes, is more easily recognized as irrelevant, while persistent low-frequency noise can confuse the model, leading to a decrease in performance for these specific noise conditions.

5 Conclusion

5.1 Review Research Findings

This research thesis introduces the application of modern deep learning techniques in the classification of space debris, assessing the performance accuracy of various architectures. A key aspect of our investigation was examining the impact of different signal processing approaches during training of the classification task.

Decibel transformation leads in many cases to improved accuracy, better generalization, and faster convergence. This improvement is credited to the compression of the dynamic range that occurs when a logarithmic scale is applied. However, there are exceptions to this trend. Architectures such as Bi-LSTM and Full-Attention Transformer achieve almost equally high accuracy levels when trained on either raw voltage signals or processed decibel signals. FullTRAN realizes this consistency using the same set of parameters, whereas Bi-LSTM requires minor modifications, such as a reduction in the learning rate. This lower learning rate leads to slower convergence, thereby demanding more training epochs to reach the same level of accuracy as that achieved with decibel-transformed signals.

Through extensive experimentation and hyperparameter tuning, we trained models across various architectures, with the best-performing models approaching an accuracy threshold of 95%. This threshold is linked to a specific misclassification issue, where tall cones and cylinders share the same RCS signature within a certain range of observation angles. Since our monostatic radar setup provides only a single perspective on the target object, addressing this issue proves challenging without modifications to the observation system.

In our robustness evaluations, we observed that models using decibel transformations before feature extraction benefit from noise reduction, improving clarity in radar distortions like clutter and saturation. Yet, FullTRAN trained on raw signals defies such a trend once more. Models excelling at classifying raw voltage signals are more robust against data occlusion since data loss is simpler to manage in raw signals. Logarithmic conversion complicates the interpretation of signals distorted by radar occlusion. Furthermore, the Full-Attention mechanism of FullTRAN demonstrates unmatched resistance to clutter from sinusoidal patterns across various frequencies, more than any other tested model.

These experimental findings highlight the potential of deep learning in space debris classification to support trajectory tracking. This advancement opens the door to safer space operations and contributes to the survival of active space infrastructures.

5.2 Possible Future Research Continuations

Despite identifying models with resilience to certain radar distortions, occlusion continues to pose a challenge for models trained on decibel signals. Artificial dropout severely disrupts the decibel scale by setting data points to near zero in raw signal samples. The following decibel conversion complicates feature extraction, proving more difficult than the initial problem of straightforward information loss.

One potential approach to preserve the integrity of the decibel scale involves bridging the data loss gaps by merging these segments. However, this strategy might introduce complications for models that rely on a consistent input size, since this would result in shorter data samples. In such cases, interpolation could serve as a viable method to address this challenge, depending on the degree of lost data.

More interesting project ideas include to enhance model resilience against specific types of radar noise by integrating data augmentation techniques that simulate various radar distortions during the training process. Furthermore, exploring the use of denoising diffusion models as a preprocessing step could provide a novel and effective approach to improve signal clarity by filtering out radar distortions before classification. These approaches represent promising directions for extending the work presented in this thesis, potentially leading to more robust and accurate space debris classification systems.

(a) Uneven rocket   (b) Rocket RCS matrix   (c) RCS 3D wrapped
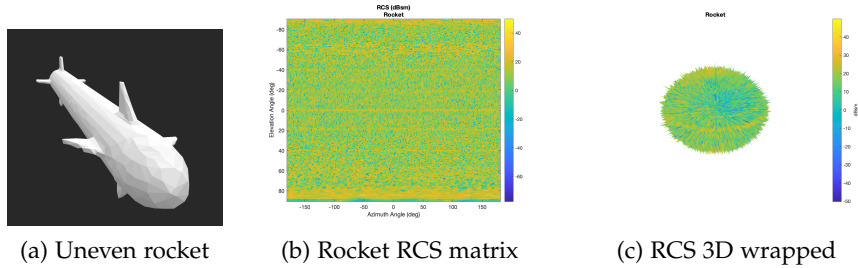
Figure 5.1: Example RCS signature of a 3D rocket with uneven surface

The achievements of our experimentation were based on the accomplishments of our simulation pipeline. It represents a great starting point for generating datasets to explore deep learning applications in space debris classification.

However, given additional time and resources, we could explore more complex simulations designed to more accurately mimic the conditions of space debris in Low Earth Orbit. Such enhancements hold the potential to fuel future research endeavors:

1. Simulating more complex objects, such as satellites, spacecraft fragments, or even rockets, to study their complex RCS signatures, as shown in Fig. 5.1.

2. Including simulations of non-fully conducting materials to understand their impact on signal detection and classification.

3. Simulating the antenna setup itself, in addition to the emitted and received radar signals, for a more comprehensive understanding of signal emission and reception.

4. Simulating the trajectory of objects flying through the antenna's field of view, resulting in more complex trajectory patterns and observation windows of varying sizes.

5. Incorporating specific noise sources in training data, such as clutter and random dropouts, to better prepare the models for real-world conditions.

List of Figures

List of Tables

Bibliography

[1] European Space Agency. *ESA Space Debris User Portal.* `https://sdup.esoc.esa.int/discosweb/statistics/`.

[2] European Space Agency. *Hypervelocity impacts and protecting spacecraft.* Tech. rep. European Space Agency, 2017.

[3] European Space Agency. *Space Debris: the Esa Approach Overview of Space Debris.* Tech. rep. European Space Agency, 2017.

[4] X. Fu, G. Liu, and M. Gao. "Overview of orbital debris detection using spaceborne radar". In: *2008 3rd IEEE Conference on Industrial Electronics and Applications, ICIEA 2008.* 2008, pp. 1071–1074.

[5] D. W. Walsh. *A Survey of Radars Capable of Providing Small Debris Measurements for Orbit Prediction.* Tech. rep. Texas A&M University, 2013.

[6] M. Ramírez-Torres, M. Ferreras, C. Hernández, C. García-de-la-Cueva, M. Barba, G. Perez-Palomino, J. A. Encinar, M. Sierra-Castaner, J. Grajal, J.-L. Vázquez-Roy, and E. Rajo-Iglesias. "Technological developments for a space-borne orbital debris radar at 94 GHz". In: *2018 IEEE Radar Conference (RadarConf18).* IEEE. 2018, pp. 0564–0569.

[7] S. Ghio and M. Martorella. "Size estimation of space debris models from their RCS measured in anechoic chamber". In: *2020 17th European Radar Conference (EuRAD).* IEEE. 2021, pp. 421–424.

[8] P. Swerling. "Probability of detection for fluctuating targets". In: *IRE Transactions on Information theory* 6.2 (1960), pp. 269–308.

[9] J. Lundén and V. Koivunen. "Deep learning for HRRP-based target recognition in multistatic radar systems". In: *2016 IEEE Radar Conference (RadarConf).* IEEE. 2016, pp. 1–6.

[10] E. Wengrowski, M. Purri, K. Dana, and A. Huston. "Deep CNNs as a method to classify rotating objects based on monostatic RCS". In: *IET Radar, Sonar & Navigation* 13.7 (2019), pp. 1092–1100.

[11] J. Mansukhani, D. Penchalaiah, and A. Bhattacharyya. "Rcs based target classification using deep learning methods". In: *2021 2nd International Conference on Range Technology (ICORT).* IEEE. 2021, pp. 1–5.

[12] Mathworks. "Matlab Radar Toolbox, AI for Radar". In: *https://de.mathworks.com/help/radar/ug/radar-target-classification-using-machine-learning-and-deep-learning.html* ().

[13] R. Battaglia, M. Ferri, V. Dainelli, F. Sarullo, and M. Demeo. "Warden: W-band advanced radar for debris early notification from ISS". In: *2003 IEEE Aerospace Conference Proceedings (Cat. No. 03TH8652)*. Vol. 1. IEEE. 2003, pp. 1–81.

[14] M. A. Richards, J. Scheer, W. A. Holm, and W. L. Melvin. "Principles of modern radar". In: (2010).

[15] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. 2019.

[16] W. Falcon. *PyTorch Lightning*. `https://github.com/PyTorchLightning/pytorch-lightning`. 2019.

[17] S. Hochreiter and J. Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[18] R. Child, S. Gray, A. Radford, and I. Sutskever. "Generating long sequences with sparse transformers". In: *arXiv preprint arXiv:1904.10509* (2019).

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[20] L. McInnes, J. Healy, and J. Melville. "Umap: Uniform manifold approximation and projection for dimension reduction". In: *arXiv preprint arXiv:1802.03426* (2018).