Extreme Gradient Boosting and Deep Learning Models for the Classification of Synthetic Space Debris Light Curves

Anne Adriano
University of Waterloo
K. Andrea Scott, Nasser Lashgarian Azad
University of Waterloo

ABSTRACT

The field of Space Domain Awareness (SDA) is becoming increasingly important, particularly amidst the growing population of space debris in low Earth orbit (LEO). Previous research has explored the classification of various types of attitudes exhibited by both defunct and operational satellites and rocket bodies, yet there is a noticeable gap in the literature concerning the analysis of attitude complexities exhibited by smaller orbital debris, such as fragments and detached satellite components, specifically. This lack of focus on debris sub-classification could be attributed to the scarcity of labeled debris data.

To address this challenge, a light curve simulation framework was created, which combined the capabilities of both Orekit for numerical orbit propagation, and Blender for physical environment modelling and image rendering. Using this simulator, light curves of different debris shapes were generated based on initial conditions defined by LEO Two-Line Elements (TLEs), and a series of physical object parameters. While there is an infinite number of shapes that orbital debris can assume following a breakup event, this study focused on photometric signatures produced by five shape classes in particular: rods, panels, cuboids, dishes, and cones.

The first task aimed to classify the light curve attitude complexity as either tumbling or stable, where tumbling objects are characterized by a rotation about two or more body axes, and stable objects are either nonrotational or rotating about a single axis. Data pre-processing for this step involved the extraction of frequency information using the Wavelet Scattering Transform (WST). Running an Extreme Gradient Boosting (XGBoost) algorithm on the extracted features for this binary classification task produced an accuracy of 90%. The second task involved the shape classification of tumbling debris using a set of deep learning models, namely a Convolutional Neural Network (CNN), a Long Short-Term Memory network (LSTM), and a fully convolutional LSTM (LSTM-FCN). The highest performer among these networks was the LSTM-FCN, achieving a test accuracy of 95%.

This study employed novel approaches to the attitude and shape classification of debris light curves, producing promising results when using a unique dataset of LEO debris. Extensions of this work will involve the implementation of attention mechanisms to further improve the performance of the deep learning models, as well as the application of more sophisticated light curve inversion techniques to extract precise attitude rates from the light curves.

1. INTRODUCTION

In recent years, the risks that the growing population of Earth-orbiting debris pose on operational satellites have increased. This has called for a significant need to study and understand the complex behaviour of space debris. Statistical models developed by the European Space Agency (ESA) estimate upwards of $40,500$ debris pieces greater than 10cm orbiting Earth as of 2024 [1], with Low Earth Orbit (LEO) containing the highest concentration of debris [2]. In order to accurately assess collision risks, it is critical to collect as much information about surrounding objects as possible.

A common approach to space object characterization is to perform light curve inversion, where a light curve refers to an object's observed brightness over time, and inversion is the process of recovering information based on this photometric signature. One publicly available light curve dataset is the Mini-MegaTORTORA (MMT) catalogue maintained by the Kazan Federal University [3], which tracks light curves in both low Earth orbit (LEO) and geosynchronous Earth orbit (GEO). This catalogue consists of over 500,000 light curves of satellites (operational and nonoperational), rocket bodies, and debris. The Space Debris Light Curve Database (SDLCD) also contains both LEO and GEO objects, but is a smaller publicly available catalogue consisting of only $1,900$ light curves [4]. Some drawbacks to both of these datasets are their lack of information on the objects' attitudes, as well as the absence of shape information specifically on the debris pieces being tracked.

This lack of information in existing publicly available databases has been addressed by creating simulation pipelines to generate the light curve data required to train machine learning and deep learning models. Previous works have chosen mathematical implementations of the Ashikhmin-Shirley and Beard-Maxwell bidirectional reflectance distribution function (BRDF) models to simulate light curves of rocket bodies and satellites [5, 6]. Badura et al. [6] perform an intensive light curve study using empirically-collected material data to physically derive inputs to their mathematical model. An alternative approach uses Blender, an open-source 3D modelling, animation, and rendering software [7], which has previously been employed for light curve simulation by Allworth et al. and Kanzler et al. [8, 9]. These studies utilize rendered images to create light curves and have shown, through comparisons with empirical light curves, that Blender-generated data capture the necessary detail for the analysis of simple shapes performed in this study. This, along with its flexibility in manipulating object dimensions and parameters, such as spin rates, surface properties, and environment settings, make Blender a more compelling option for our study than mathematical implementations.

The debris, rocket body, and satellite light curves simulated by Linares et al. [5] were classified based on shape using a convolutional neural network (CNN), achieving an accuracy of 98%. With the growing popularity and success of CNNs in the Space Domain Awareness (SDA) community, they have also been applied to attitude classification [6], alongside long short-term memory (LSTM) networks [10, 11, 12]. In the majority of cases, deep learning models have shown to outperform traditional machine learning models on time series classification tasks because of their ability to learn features and model nonlinear relationships within the data.

To aid the classification models, [13, 11, 10] use variations of the Wavelet Transform to extract features from light curves for classification tasks. Wavelet-based frequency analysis methods preserve time information, unlike the Fast Fourier Transform, which assumes a signal is infinite and does not capture temporal details. Dao et al. apply the Wavelet Transform to obtain frequency information from GEO light curves, and use this information to distinguish between satellites and rocket bodies [13]. Alternatively, [11, 10] apply the Discrete Wavelet Transform (DWT) and the Wavelet Scattering Transform (WST), respectively, to LEO light curves for feature extraction. While the DWT and WST both feed the signal through a series of low pass filters for frequency component analysis, the wavelet convolutions and modulus operations applied in the WST produce features that are both translation and deformation invariant [14]. Consequently, Qashoa and Lee argue that the latter technique outputs richer features without the need for additional tests for periodicity and aliasing [10].

Upon distinguishing a fragmented space debris from satellites and rocket bodies, further classifying the debris based on rotational and shape characteristics, in addition to considering orbital perturbations, would facilitate more accurate trajectory propagations. This debris sub-classification has yet to be explored via light curve data. Deriving this additional information would enhance our understanding of potentially uncatalogued objects, allowing for more informed collision risk assessments.

This study aimed to combine the methodologies from previous works for both data simulation and light curve classification. The first step involved the generation of a unique dataset of labelled debris light curves. The shape and spin rate labels in the synthetic data offered an advantage over existing publicly available datasets by allowing for the classification based on these characteristics. As such, the second step in this work involved the attitude classification of the light curves using an Extreme Gradient Boosting (XGBoost) model in conjunction with the WST, followed by the shape classification of tumbling debris using a series of deep learning models.

Section 2 outlines our data simulation framework, followed by a description of the employed classification methods in Section 3. Results and a discussion of the findings from this research are summarized in Sections 4 and 5, respectively.

2. DATA SIMULATION

2.1 Object Initialization

The simulator used to generate the light curve data was written entirely in Python and combines the capabilities of both Orekit for numerical orbit propagation and Blender for the space environment set up and image rendering. Figure 1 shows an overview of the simulation process, which is repeated for each light curve. It should be noted that this process is similar to that used by Allworth et al. [15], which also involves an orbit propagator in conjunction with a physical environment setup in Blender to create rendered images of a space object from a ground station.

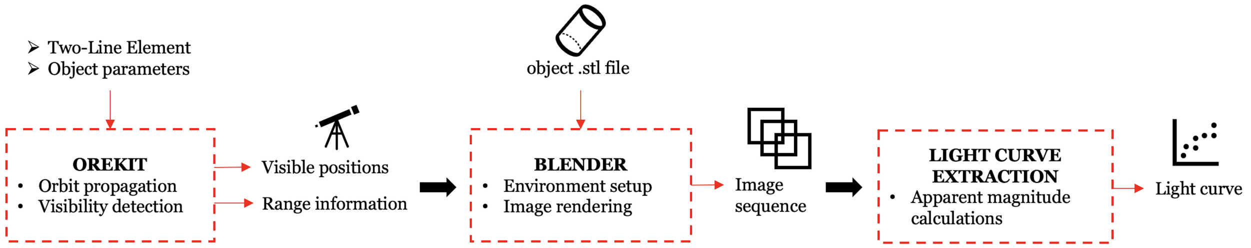


Fig. 1: Light curve simulation framework involving the integration of Orekit, Blender, and a combination of apparent magnitude calculations from [5, 16].

Prior to the data simulation process, 3D models representing various shapes of space debris were created. Our shape bank consisted of truncated cones (frustums), dishes, plates, cuboids, and rods. These were designed to represent pieces of satellites that may break off during breakup events, which might include thruster nozzles, antennas, and solar panels. As such, realistic dimensions were sourced from publicly available data on existing satellites and dimension specifications of their components [17, 18, 19]. To create a more comprehensive dataset, three differently proportioned variations of each shape were created, resulting in a total set of 15 unique objects. Each object was assigned a cross sectional area, mass, and drag coefficient [20], which are used when calculating the atmospheric drag in the orbit propagation step of the simulator.

A list of 20 Two-Line Elements (TLEs), corresponding to existing LEO debris, was sourced from *spacetrack.org* [21]. Table 1 summarizes orbit information extracted from five of the 20 TLEs used to generate the light curves in this study.

Table 1: Sample LEO TLE information sourced from *spacetrack.org* used for orbit initialization. Orbital characteristics shown from left to right are the semimajor axis, inclination, eccentricity, right ascension of ascending node, argument of perigee, and mean anomaly.

Object	NORAD ID	a [km]	i [°]	e	Ω [°]	ω [°]	M [°]
Oscar 23 DEB	33395	7527.97	90.31	0.0186	354.41	324.20	99.95
NOAA 11 DEB	37241	7182.58	98.91	0.0021	252.34	351.79	8.29
Falcon 9 DEB	52893	6878.90	52.99	0.0008	89.72	140.84	219.32
Ariane 1 DEB	17129	7185.22	98.37	0.0033	137.52	21.67	150.36
Iridium 33 DEB	33850	7139.97	86.34	0.0012	269.57	202.67	226.95

Apart from the shape of the object, this work also aims to classify the object's attitude complexity. As such, the initialization step also sets the object's attitude regime as either tumbling or stable. A tumbling object corresponds to a rotation about 2 or 3 of its body axes, and a stable object is considered either nonrotating or rotating about a single body axis. This is followed by the randomization of which axes will be rotating depending on the attitude regime. The period of rotation about a given axis is randomly sampled between 2 and 30 seconds.

2.2 Numerical Orbit Propagation: Orekit

Orekit is a comprehensive Java library for space flight dynamics and astronomy, offering tools for satellite orbit propagation and space mission analysis [22]. To facilitate its use in this research, the Python wrapper for Orekit was used, allowing for integration with our Python-based framework. This API served as the orbit propagator for each object, where the sampled TLE determined the initial conditions for the numerical orbit propagation process. To incorporate the effects of atmospheric drag, solar radiation pressure, and third body attractions from the sun and moon, the propagator also accounts for the object's mass, cross sectional area, drag coefficient, and reflectivity coefficient.

A ground field of view detector is used alongside the orbit propagator in order to detect the region of the orbit in which the object is visible from the ground station, which was set to have the same longitude, latitude, and elevation as the MMT telescope [3]. The visibility detector checks a circular field of view directly above the ground station with a 60 degree radius, and considers factors such as sun elevation and eclipses. Figure 2 shows the orbit propagation of an object using the Ariane 1 DEB TLE from Table 1.

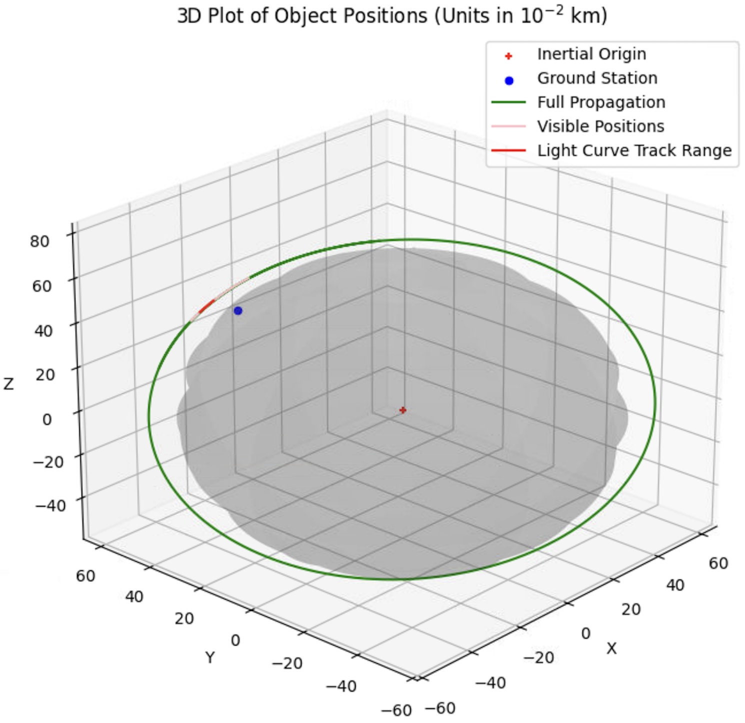


Fig. 2: Numerical orbit propagation results generated by Orekit using an Ariane 1 debris TLE and object parameters corresponding to a rod. Visible range and light curve region are also shown along the orbit in pink and red, respectively.

The plot illustrates the orbit of an object around Earth, with markers positioned at Earth's center (red) as well as at the location of the ground station (blue). The axes correspond to the Earth-Centered Inertial (ECI) frame, where the x-axis aligns with the direction of the vernal equinox, the y-axis lies within the equatorial plane, perpendicular to the x-axis, and the z-axis points along Earth's north pole. Figure 2 also shows the object's full orbit (green), the region of visibility (pink), as well as the region during which the light curve for this object was recorded (red); this corresponds to a 60 second long track. The object in this track is located at an altitude of approximately $1,400$km.

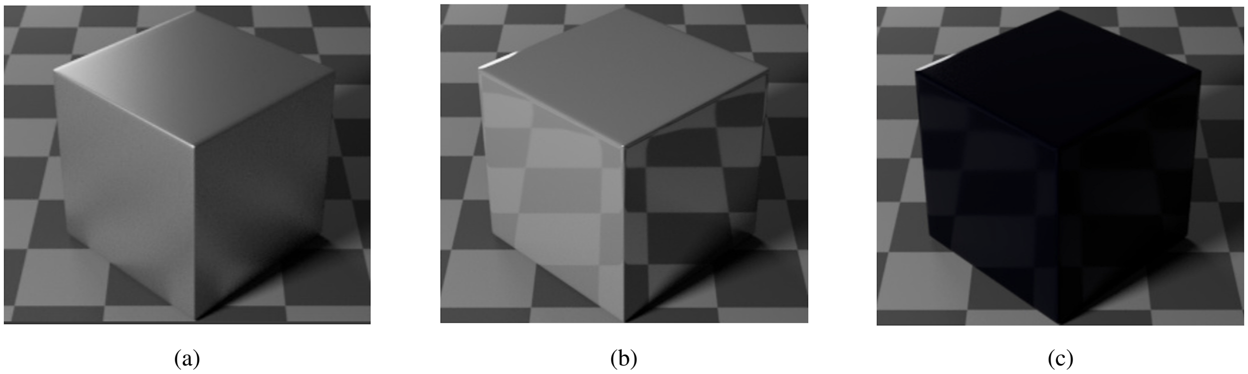
2.3 Light Curve Simulation: Blender

Among the object's positional information, Orekit outputs other necessary data for setting up the physical environment in Blender, a powerful open-source 3D graphics and animation software previously used for light curve simulation [15, 23]. Blender plays a critical role in this light curve simulation pipeline by modeling the physical space environment and rendering the required images.

The positions of the ground station and the Sun are fixed for the duration of the light curve track. These positions are derived from historical data corresponding to the time of the light curve observations to ensure that the simulation accurately reflects the observational conditions during the epochs of interest. The distances provided by Orekit in meters are scaled appropriately to fit in the Blender scene, and are accounted for in the light curve extraction step.

The solar irradiance is set to 1062 W/m², aligning with the standard solar constant. This value represents the average solar energy received per unit area at the Earth's surface, which is essential for simulating realistic lighting conditions and accurate reflections in the rendered images. The Earth is also modelled and centered at the scene's origin to account for earthshine effects.

The 3D model of the object is then imported into the environment and placed at the position corresponding to a randomized start index within the list of visible positions. This is followed by the assignment of physical parameters, which are associated with one of three possible materials that an object can assume in this study: diffuse, specular, and solar. Although the former two finishes are not completely diffuse or specular, they were modeled to resemble the surface finishes of brushed aluminum and aluminum foil reflectors, respectively. The solar material was designed to simulate the colour and finish of a solar panel. Figure 3 shows renderings of these materials in Blender.



(a) (b) (c)

Fig. 3: Three types of material finishes used in the light curve simulator: (a) Diffuse, (b) Specular, and (c) Solar.

Each object is assigned a uniform material composition, where dishes can only be specular (to model satellite reflectors), panels can be assigned any of the three materials, and all other objects (cuboids, rods, and cones) can only be diffuse. Table 2 summarizes the parameter settings that were used to configure the materials shown in Figure 3.

Table 2: Blender parameters corresponding to the three types of surface finishes shown in Figure 3.

Finish	R	G	B	Roughness	Metallic	Diffusivity	Reflectivity	IOR
DIFFUSE	0.6	0.6	0.6	0.4	1.0	0.6	1.2	1.5
SPECULAR	0.6	0.6	0.6	0.05	0.8	0.1	1.7	2.0
SOLAR	0.002	0.001	0.012	0.1	0.8	0.6	0.5	1.5

The object is then given a random set of Euler angles about its body axes to define its initial orientation. From there, the object is spun according to the spin rates assigned to the object during the initialization step. The Nyquist theorem states that to capture a signal without aliasing, the sampling frequency must be at least 2x the maximum frequency of the signal [24]. In this study, the maximum possible spin frequency about a given axis is 0.5Hz, indicating that the sampling frequency would need to be at least 1Hz to accurately capture the rotation of the fastest spinning object. The chosen sampling frequency was 5Hz, corresponding to one image taken every 0.2 seconds.

To speed up rendering times, the camera at the ground station location tracks the object of interest and zooms in by 80%, which allows the software to focus only on the required information to be rendered, and limits the amount of empty space in the aperture. Each light curve in the dataset was set to record 300 measurements, which is equivalent to 60 seconds.

2.4 Light Curve Extraction

The method used to construct a light curve using the pixel counts derived from the rendered images was taken from [15], which involved techniques for extracting both the apparent magnitudes via the flat facet model defined in [5] and the object's instrument magnitudes at each time step [16].

The apparent magnitude of a target object can be determined by a relationship that considers the photon flux of that object (f_{target}) along with the photon flux (f_{ref}) and known magnitude (m_{ref}) of a distant reference star [15],

$$m_{target} = m_{ref} + 2.5log_{10}(f_{ref}) - 2.5log_{10}(f_{target}). \tag{1}$$

In scenarios where the reference object is not a star and its apparent magnitude (m_{ref}) is unknown, the flat facet model derived from the Ashikhmin-Shirley BRDF in [5] describes a method based on the apparent magnitude of the sun (-26.7) and the discretization of the reference object's visible surfaces into a series of flat facets. The contribution of a single flat facet, j, to the reference object's overall apparent magnitude is defined by:

$$m_{ref,j} = -26.7 - 2.5log_{10}\left[\frac{(\hat{u}_{n,j}\cdot\hat{u}_{s,j})\,\rho_{tot,j}}{d^2_{ref,j}}\right], \tag{2}$$

where $\rho_{tot,j}$ is the total diffuse reflectance of the facet's material and $d_{ref,j}$ is the distance between the camera and the facet. The dot product between the facet's normal vector and the vector from the facet to the sun, $(\hat{u}_{n,j}\cdot\hat{u}_{s,j})$, quantifies how much facet j is rotated in the sun's direction, away from the observer. This value is inversely proportional to the amount of sunlight reflected towards the observer.

The simulation in this study creates one reference object for each light curve. This 1m by 1m flat plate is placed $10,000$km above the ground station and oriented such that the camera is normal to the plate's surface. The contribution of each facet j given by Equation 2 is typically summed over all J facets, but in this case, the reference object's 1m^2 surface area represents the only visible facet required to calculate the reference apparent magnitude, m_{ref}.

In certain cases, the exact photon flux is also unknown, but brightness information is available via images taken by a telescope. The instrument magnitude described in [16] can be applied in this scenario, which provides an alternative means of quantifying the magnitude using the image pixel count, B, instead of the photon flux of any object:

$$I = -2.5log_{10}\left(\frac{B - AS_{sky}}{t_{exp}}\right), \tag{3}$$

where A is the aperture area (in pixels), S_{sky} is the sky signal per pixel, and t_{exp} is the exposure time of the image, in seconds. Since S_{sky} in Blender is 0 [15], the logarithm in Equation 3 simplifies to $\frac{B}{t_{exp}}$. The logarithm terms in Equation 1 can be replaced with the corresponding instrumental magnitudes, yielding the following general equation for the apparent magnitude of the target object at each time step, i

$$m_{target,i} = I_{target,i} + m_{ref} - I_{ref}. \tag{4}$$

To account for the scaled distances and in Blender, a correction factor is applied to the instrumental magnitude of the target object, which gives the following

$$I_{target,i} = -2.5log_{10}\left(\frac{B_i}{t_{exp}\left(\frac{d_i}{d_{ref}}\cdot scale\right)^2}\right). \tag{5}$$

In this equation, d_i is the real world distance between the observer and the target object and d_{ref} is the distance between the observer and the reference object. Plugging all variables into Equation 5 and simplifying, we obtain,

$$m_{target,i} = -26.7 - 2.5log_{10}\left[\frac{(\hat{u}_n\cdot\hat{u}_s)\,\rho_{tot,i}}{d_i^2 scale^2}\frac{B_i}{B_{ref}}\right]. \tag{6}$$

It is worth noting that both t_{exp} and d_{ref} cancelled out during the simplification of Equation 5, suggesting that the exposure time of each image and the distance at which the reference object is placed with respect to the observer do not affect the final apparent magnitude at any given time step. Applying this overall equation using pixel counts from the 300 rendered images of each track produces a time series consisting of apparent magnitude values. The light curve generated from the orbit information displayed in Figure 2 is shown in Figure 4.

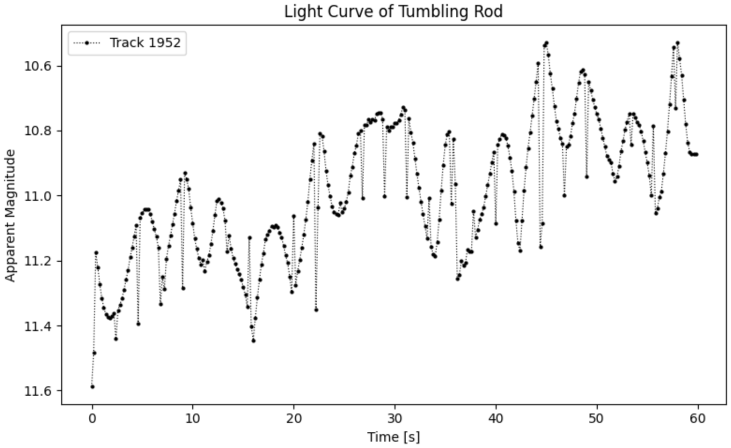
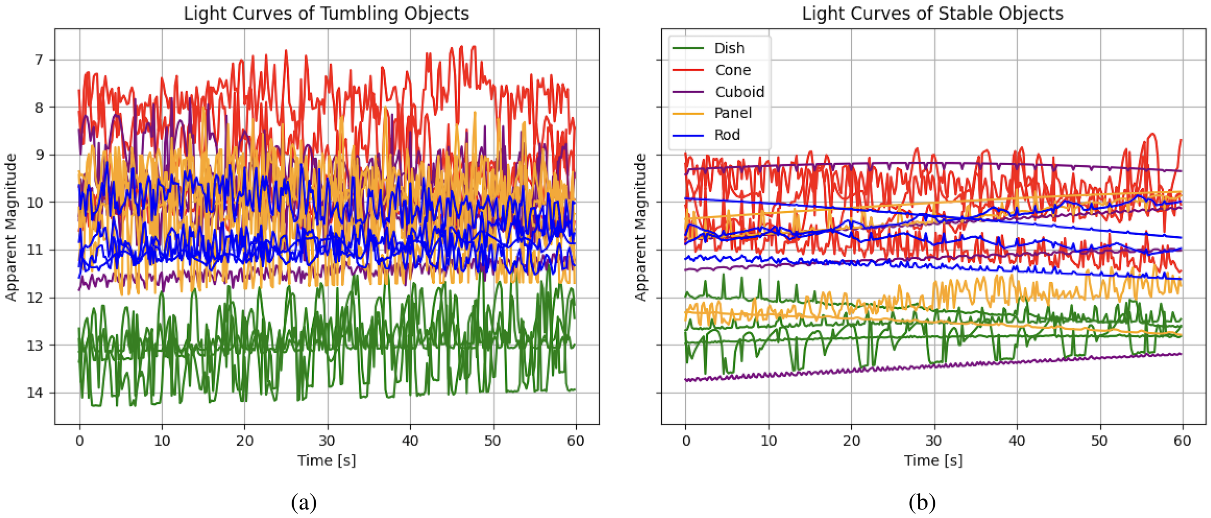


Fig. 4: Generated light curve of a rod rotating about two axes, corresponding to the red light curve region shown in Figure 2. Inverted y-axis for intuitive visualization of brightness variations, reflecting the inverse relationship with apparent magnitude values.

The generation of a light curve in our simulation pipeline takes, on average, 16.6 minutes. In this plot, the y-axis has been inverted to reflect the inverse relationship between brightness and apparent magnitude, making the data easier to interpret. The generated light curve is of a rod tumbling with a rotation about two axes. While this object was assigned a primarily diffuse surface finish, some specular behaviour is still present. Sharp dips in the object's brightness may be due to the object's rotation and how certain orientations are associated with a lower visibility of these specular areas from the point of view of the observer. Figure 5 provides a visual for simulated light curves of different attitude regimes with color coded object shapes.



(a)

(b)

Fig. 5: Plots of tumbling (a) and stable (b) light curves with shape labels, showing four different representative light curves for each shape. The tumbling objects exhibit chaotic and highly variable brightness, with frequent and significant fluctuations in apparent magnitude. In contrast, the stable objects demonstrate smoother and more consistent light curves, reflecting steadier brightness and less erratic behavior.

A total of 6,500 labelled debris light curves were generated for this classification study, where both the attitude regime and shape of each light curve were chosen from a uniform distribution to ensure that the classes would be well balanced for the investigated classification tasks. As expected upon visual inspection, stable objects (either nonrotating or rotating about a single axis) exhibit less variability in fluctuations than tumbling objects due to a steadier orientation relative to the point of observation.

Light curves of dish objects are, in general, noticeably dimmer than the other light curves based on Figure 5. This can be attributed to both the concave shape of satellite dishes and the fact that this object class was simulated only using the specular material type. Sunlight hitting the dish will only reach the observer if the angle between the sun and the surface normal equals the angle between the observer and the surface normal. In other words, the visibility of a specular object is highly dependent on its rotation relative to both the position of the sun and the observer. The concavity also makes the dish's visibility more selective and dependent on precise geometrical alignment. In contrast, diffuse material types reflect light in all directions, making a diffuse object's geometry less restrictive for visibility from the point of observation.

3. CLASSIFICATION MODELS

3.1 Extreme Gradient Boosting

XGBoost is a machine learning ensemble method, which combines the capabilities of a series of decision trees. The model leverages regularization, pruning, shrinkage, as well as column and row subsampling to iteratively improve performance. Each subsequent tree aims to correct the residuals or errors made by the previous trees, and they are also penalized based on tree complexity. Several hyperparameters can be tuned to optimize performance, making XGBoost a highly scalable model for any regression or classification task. For a comprehensive overview of the model, readers are encouraged to consult [25].

The first classification task used XGBoost to distinguish between tumbling and stable light curves, where tumbling objects are defined by a rotation about two or three body axes, and stable objects are either nonrotating or rotating about a single axis. The model is trained over 2000 epochs, with an early stopping criterion set to terminate training if there is no improvement in the negative log likelihood loss for 50 consecutive epochs. During each epoch, the model randomly selects a subsample comprising 80% of the entire dataset to grow each tree. To mitigate overfitting, the maximum tree depth is limited to 10 levels, and at each level, 80% of the features are randomly chosen for consideration. From these, 50% are randomly selected for splitting at each node. This randomness in feature selection helps increase the diversity of trees and enhances the model's robustness. XGBoost has shown promising results in other related time series classification tasks [26, 27], making it a compelling method to apply as a novel approach to light curve classification.

3.2 Convolutional Neural Network

A CNN is a type of neural network that learns hierarchies of features in the input data using convolutional filters. Otherwise referred to as kernels, these filters are able to scan across one-dimensional data in a 1D-CNN to detect local patterns, periodicity, and temporal dependencies, among other characteristics. This hierarchical feature extraction is followed by pooling layers, which reduce the spatial dimensions of the data, thereby lowering computational requirements and promoting feature invariance. The extracted features are then fed into fully connected (FC) layers that act as a classifier, making predictions based on the learned features. The training of a CNN uses backpropagation, which adjusts the filters and weights in response to the error between its predictions and the actual labels.

Since the classification of object shapes focused only on tumbling objects, the dataset was limited to the $3,200$ tumbling light curves, where the training, validation, and test sets represented 80%, 10%, and 10% of the tumbling light curve data.The 1D-CNN architecture utilized in this study was similar to that employed by Furfaro et al. [5], which has been proven to perform well on their simulated GEO light curve data for the classification of fragments, rocket bodies, regular polygon prisms, and rectangular cuboids.

This architecture consists of three convolutional layers with 64, 32, and 64 filters of sizes 32, 12, and 6, respectively. Each layer is followed by rectified linear unit (ReLU) activation, dropout regularization of 0.2, and max pooling with a kernel size of 1x4. This is followed by a FC layer containing 100 neurons, 0.5 dropout regularization and a final FC layer to align the model's output with the number of shape classes. This differs slightly from the two FC layers with 500 and 100 neurons, and softmax function used by Furfaro et al. [5]. The model is trained for 2000 epochs, with early stopping criteria set to 100 epochs without improvement. The network processed the data in batch sizes of 128, applying cross-entropy loss for classification and stochastic gradient descent with a learning rate of 0.01 and a momentum of 0.9.

3.3 Long Short-Term Memory Network

LSTM networks are a specialized form of recurrent neural networks (RNNs) designed to capture temporal dependencies in sequential data. Traditional RNNs suffer from the vanishing gradient problem, where gradients of the loss function shrink exponentially over time. The recursive nature of RNNs also requires the recomputation of the internal state for every input, making it difficult for these networks to track and learn long-range dependencies [28]. LSTM networks address these issues through their unique architecture that includes memory cells and gating mechanisms.

An LSTM consists of a series of connected LSTM units, each containing a cell state that is modulated by three primary gates: the input gate, the forget gate, and the output gate. The cell state serves as the network's long-term memory, allowing for the preservation of relevant information across many time steps. The gates regulate the flow of information into and out of the cell state, ensuring that relevant information is added or retained, and that nonessential information is discarded. This structure allows LSTMs to effectively learn and remember context over long sequences, making them powerful for a variety of sequential data tasks.

The LSTM model in this study consisted of four stacked LSTM layers, each with 64 hidden units. The dropout rate was set to 0.1 and the learning rate was 0.006. The model was trained over $2,000$ epochs with early stopping after 100 rounds without improvement, using batches of 128 samples and cross-entropy loss.

3.4 Fully Convolutional Long Short-Term Memory Network

A fully convolutional LSTM (LSTM-FCN) is an advanced neural network architecture designed to handle sequential data by combining the strengths of convolutional layers and LSTM units. This hybrid approach is particularly effective for time-series analysis, where the goal is to capture both local patterns and long-term dependencies in the data [29]. This network architecture is shown in Figure 6.

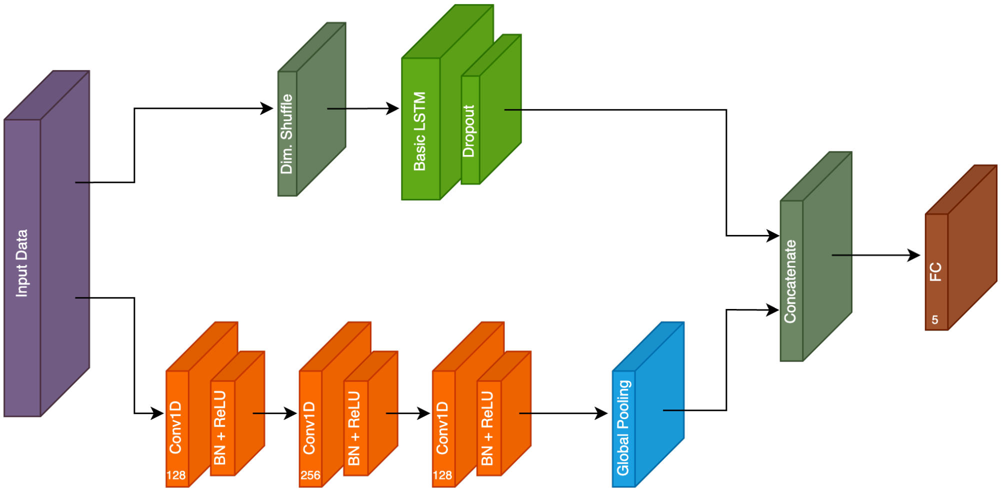


Fig. 6: A hybrid model architecture combining an LSTM and a CNN, for time series classification. The input data is processed along two parallel paths: the top path involves reshaping the input data and feeding it into an LSTM followed by a dropout layer, and the bottom path employs a three-layer CNN and global average pooling. The outputs from both paths are concatenated and fed into an FC layer to produce the final class predictions.

A dimension shuffle is performed on the input data to transform it into a multivariate time series with a single time step, which was shown in [29] to enhance the architecture's performance. The LSTM block used for this study was the same as the LSTM network described in Section 3.3. A dropout rate of 60% was applied to the LSTM block output to reduce overfitting.

The three convolutional layers use 128, 256, and 128 filters, of sizes 8, 5, and 3, respectively. Each layer is followed by batch normalization and ReLU activation. Global average pooling is then applied to the output of the final convolutional layer to reduce the number of parameters prior to concatenation with the LSTM results. Following the concatenation step, an FC layer to produce the final classification results.

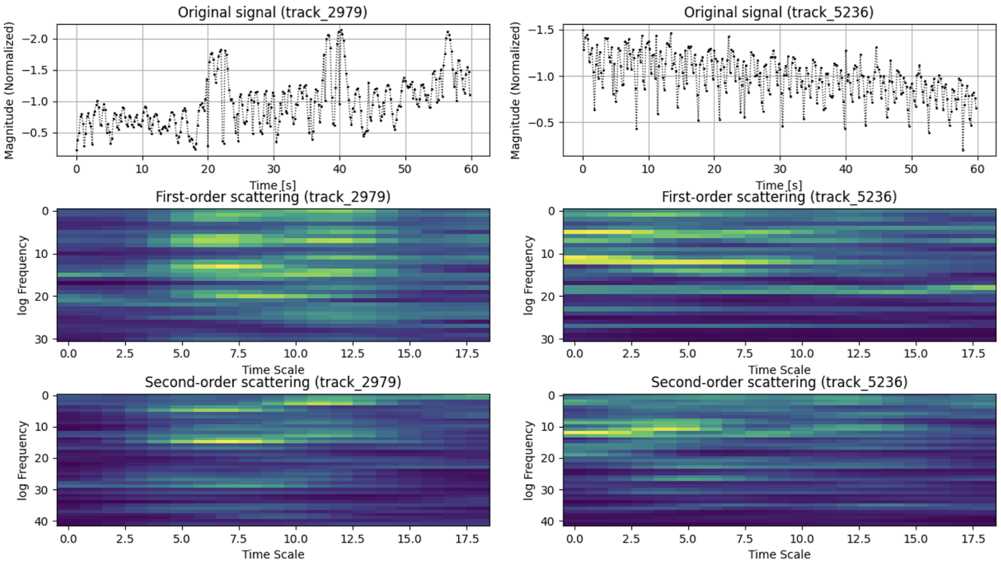
4. RESULTS

4.1 Attitude Classification

To classify the light curves based on attitude with XGBoost, we first employ the WST for feature extraction. The primary advantage of this method lies is its ability to extract prominent frequency components while preserving valuable time-domain information. This is particularly useful in the context of light curve analysis, where each light curve exhibits signatures that reflect the object's rotation profile. The transformation from apparent magnitude values to scattering coefficients is achieved by iterating over three main steps [14].

The first step of each layer (or iteration), is to convolve the input signal with wavelets of different scales defined by inputs J and Q. J determines the number of scales used in the transform, where the maximum scale is 2^J. This parameter is inversely proportional to the temporal resolution of the resulting scattering coefficients. The Q parameter is the number of wavelets per octave, which is proportional the frequency resolution of the coefficients. The second step applied a modulus operator to the resulting signals, containing both complex and real components, to obtain the absolute value. Finally, low-pass filtering is applied to perform local averaging over a window of points in the signal and extract the first order coefficients. Convolving the signals from step two with the wavelets in the subsequent filter bank, followed by modulus and low-pass filtering, extracts the second order coefficients. The coefficients produced at each level of the transform are invariant to both time-warping and translations to a certain degree dictated by the scaling factor J.

Scattering coefficients are commonly expressed in the form of spectrograms, where the x-axis represents the time resolution, and the y-axis is the log frequency. Each coefficient is an amplitude value associated with a certain time and frequency window. Examples of the wavelet scattering transform, as applied to a tumbling cone and a stable cone respectively, are shown in Figure 7.



(a) (b)

Fig. 7: Light curves and corresponding wavelet coefficient spectrograms of cones for tumbling (a) and stable (b) attitude profiles. The top row shows the z-normalized raw light curves, followed by the first and second order WST coefficient spectrograms in the middle and bottoms rows, respectively.

The spectrograms in Figure 7 were created from WST coefficients corresponding to a scale index of 4 and 16 wavelets per octave for the first order coefficients. By default, the second order coefficients are generated using a filter bank with one wavelet per octave. These parameters corresponded to the shown frequency and time resolutions, which summarize the strength of frequency components present in different time windows within the light curves.

Figure 7a represents a cone rotating about three axes, showing a more erratic and complex behavior than Figure 7b, which is a cone rotating about only one axis. In the first order coefficients given by the middle row, the spectrogram of the tumbling cone shows a wide range of coefficient amplitudes across all frequencies, while also capturing how the strength of the frequency components vary with time. The stable object shows more uniform patterns with fewer variations over time and across the frequencies. This suggests that the WST was able to capture the difference in rotational complexity between the two signals. The second order coefficients in the bottom row quantify interactions between different frequency components that were not captured by the first filter bank, further characterizing the modulations in object rotation.

Transforming the raw light curve data into wavelet coefficients augmented the feature space for this classification task from the original 300 apparent magnitude measurements (equivalent to a 60 second light curve track) to a total of $1,387$ coefficients. Beyond extracting features using the WST, the XGBoost classifier was also tested on a different number of the extracted coefficients. The final classification results are shown in Table 3.

Table 3: XGBoost binary attitude classification results on raw light curves and WST coefficients.

Model	Features	Test Accuracy	Validation Accuracy	Precision	Recall	F1
XGBoost	300	0.778	0.758	0.78	0.78	0.78
WST-XGBoost (All Coeff.)	1387	0.900	0.897	0.90	0.90	0.90
WST-XGBoost (Order 1 Coeff.)	570	0.876	0.897	0.88	0.88	0.88

The highest performing model was the XGBoost classifier that used all $1,387$ scattering coefficients, producing a test accuracy of 90%. This represented a 12% higher accuracy than the same XGBoost model's performance using raw light curves. Although the WST coefficients offered an effective set of frequency features for this task, it was important to assess whether all $1,387$ features were necessary to achieve meaningful results. By eliminating the second order wavelet coefficients, the feature space was reduced to 570, keeping only the coefficients produced by the first set of wavelet convolutions. Doing so produced a test accuracy of 88%, demonstrating that they alone were sufficient for a meaningful improvement over raw light curve data.

4.2 Shape Classification

The three deep learning models described in Section 3 were applied to the generated light curves, where each light curve consisted of 300 apparent magnitude measurements. The objective of this task was to classify tumbling objects based on their shapes, which included rods, truncated cones, panels, dishes, and cuboids. A total of 3,200 light curves were available for this task, with 80% of the data allocated for training, 10% for validation, and 10% for testing. The results obtained from these models are shown in Table 4.

Table 4: Shape classification results of the deep learning models explored in this study based on raw light curve data.

Model	Test Accuracy	Validation Accuracy	Precision	Recall	F1
1D-CNN	0.788	0.785	0.79	0.79	0.79
LSTM	0.886	0.905	0.89	0.89	0.89
LSTM-FCN	0.946	0.880	0.95	0.95	0.95

The CNN architecture proposed by Linares et al. produced a 79% accuracy when applied to the data generated for this study. The LSTM produced a 89% test accuracy on the generated dataset, which was higher than that produced by the 1D-CNN by 10%. Leveraging a CNN's ability to detect local patterns along with the temporal patterns extracted by this LSTM network, the LSTM-FCN model achieved a test accuracy of 95%.

5. DISCUSSION

This study aimed to classify the synthetic debris light curve data based on attitude complexity and shape. A number of classification methods were implemented, namely, an XGBoost classifier for attitude classification, as well as a CNN, an LSTM and an LSTM-FCN for shape classification.

The XGBoost classifier's higher performance on the WST coefficients compared to raw light curve data can be attributed to the ability for the WST to extract detailed time-frequency features related to the rotation of the objects. Consequently, this extracted set of features was shown to be advantageous to a classification task involving the object's attitude complexity. Increasing the number of features per sample from 300 to $1,387$ might also be a contributing factor to the higher accuracy, as this provides the model with more information for making predictions. Reducing the feature set to the 570 first order coefficients produced results that were still 10% higher than the model's performance on the raw light curves. This indicated that the coefficients extracted in the first level of the WST contained enough discriminating information on the light curves to offer a meaningful improvement over time series of apparent magnitudes.

A similar data pre-processing approach was taken by Qashoa and Lee [10], which used t-Distributed Stochastic Neighbor Embedding (t-SNE) on WST coefficients as a non-linear dimensionality reduction technique. XGBoost's results using all WST coefficients achieved a 3% higher accuracy than the Support Vector Machine (SVM) employed for the classification of their embedded wavelet coefficients. The results obtained by the XGBoost model were also only 2% lower than their LSTM accuracy on the same task. This shows that XGBoost, which is based on decision trees, can produce comparable results (on light curve classification tasks) to deep learning models built to learn hierarchies of features that are often missed by classical machine learning models.

Linares et al. [5] applied a similar CNN architecture used in this study to their simulated GEO light curves for the classification of fragments, rocket bodies, regular polygon prisms, and rectangular cuboids, obtaining a 98% accuracy. The 13% lower accuracy obtained by the same model on the dataset simulated in this study can be attributed to a limited amount of training data. Linares et al. trained on $5,500$ more light curves compared to this model, making for a more robust model that is less likely to overfit. It should also be noted that their simulated training data were near geosynchronous orbit, whereas the data it this study used LEO initial conditions, making it difficult to make direct comparisons due to the nature of the data itself. Employing an LSTM to the simulated data produced a notable improvement over the CNN, which could be an indication that the data exhibit important discriminating temporal patterns that the CNN was unable to capture, but the long-term memory of the LSTM was able to identify.

Although the test accuracy produced by the LSTM-FCN was high, the 7% difference between test and validation accuracies suggests that the model's weights generalized more effectively to the test dataset. This indicated that the test data more closely resembled the training data compared to the validation data. A method to combat this bias in future investigations would be to perform cross validation and to average the test accuracies across several folds.

6. CONCLUSION

Given the growing population of space debris, it is increasingly likely for a space asset to encounter new or uncatalogued debris. Additionally, when debris pieces are identified and distinguished from satellites or rocket bodies, it is critical to further characterize the object being observed for collision avoidance. As such, it was of interest to investigate and sub-classify light curves of LEO debris, specifically, by employing machine learning and deep learning models for the first time. This was done using a dataset of $6,500$ labelled light curves, generated using the combined capabilities of Orekit and Blender.

First, the light curves were categorized based on attitude complexity, as either tumbling or stable, based on extracted wavelet scattering coefficients. The highest classification accuracy achieved by the XGBoost model on this task was 90%. This study was the first to employ XGBoost for the classification of resident space object (RSO) light curves, and results using this model were improved by 12% when combined with WST features instead of raw light curve data.

A novel approach was also taken for the second classification task, which employed an LSTM-FCN to light curve classification for the first time. A CNN and an LSTM were tested separately for the same task. These models classified the tumbling objects into five different shape categories (rods, truncated cones, panels, dishes, and cuboids), where the highest performer was the LSTM-FCN achieving a 95% accuracy.

Due to time constraints, the size of the generated dataset was limited. Future work will involve a larger dataset, with a wider range of shapes that the debris pieces can assume. For example, it would be beneficial to involve shapes with imperfections to model light curves of debris that more closely represent fragmented or damaged objects. Likewise, incorporating objects that have facets of different materials instead of one type of material per object would help in diversifying the dataset and making it more realistic.

Extensions of this work might include a study to reconstruct the assigned spin rates, similarly to methods described in [30]. Regarding the classification models, improvements will be made by (1) optimizing the XGBoost and LSTM hyperparameters using Bayesian Optimization and (2) introducing an attention mechanism as described in [29] to help improve the LSTM-FCN shape classification results.

7. ACKNOWLEDGEMENTS

We would like to extend our gratitude to Columbiad Launch Services based in Ontario, Canada, for their invaluable support and collaboration throughout this research. Special thanks to Haroon Oqab and George Dietrich for their insightful contributions and guidance. Their expertise and dedication have been instrumental in shaping our work and driving it to fruition.

8. REFERENCES

[1] European Space Agency. Space debris by the numbers, 2024. Accessed: 2024-07-22.

[2] J. C. Jones and M. Strong. Modelling small orbital debris remediation in low earth orbit. In *The Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)*, 2022.

[3] S. Karpov. Massive photometry of low-altitude artificial satellites on mini-mega-tortora. *Revista Mexicana de Astronomía y Astrofísica (Serie de Conferencias)*, 28:112–113, 2016.

[4] J. Šilha, S. Krajčovič, M. Zigo, J. Tóth, D. Žilková, P Zigo, L. Kornoš, J. Šimon, T. Schildknecht, E. Cordelli, A. Vananti, H. K. Mann, A. Rachman, C. Paccolat, and T. Flohrer. Space debris observations with the slovak ago70 telescope: Astrometry and light curves. *Advances in Space Research*, 65(8):2018–2035, 2020.

[5] R. Furfaro, R. Linares, and V. Reddy. Space objects classification via light-curve measurements: Deep convolutional neural networks and model-based transfer learning. In *The Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)*, 2018.

[6] G. P. Badura, C. R. Valenta, and B. Gunter. Convolutional Neural Networks for Inference of Space Object Attitude Status. *Journal of the Astronautical Sciences*, 69(2):593–626, 2022.

[7] Blender Foundation. *Blender - a 3D modelling and rendering package*. Blender Foundation, Amsterdam, Netherlands, 2024. Version 4.2.0.

[8] J. Allworth, L. Windrim, J. Bennett, and M. Bryson. A transfer learning approach to space debris classification using observational light curve data. *Acta Astronautica*, 181:301–315, 2021.

[9] R. Kanzler, J. Silha, T. Schildknecht, B. Fritsche, T. Lips, and H. Krag. Space debris attitude simulation - iota (in-orbit tumbling analysis). In *Proceedings of 70th International Astronautical Congress (IAC)*, 2015.

[10] R. Qashoa and R. Lee. Classification of low earth orbit (leo) resident space objects' (RSO) light curves using a support vector machine (SVM) and long short-term memory (LSTM). *Sensors*, 23(14), 2023.

[11] K. Balachandran and K. Subarao. Classification of residant space objects by shape and spin motion using neural networks and photometric light curves. In *European Conference on Space Debris*, 2021.

[12] G. P. Badura, C. R. Valenta, L. Churchill, and D. A. Hope. Recurrent neural network autoencoders for spin stability classification of irregularly sampled light curves. In *The Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)*, 2022.

[13] P. Dao, K. Haynes, S. Gregory, J. Hollon, T. Payne, and K. Kinateder. Machine classification and subclassification pipeline for geo light curves. In *The Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)*, 2019.

[14] J. Anden and S. Mallat. Deep scattering spectrum. *IEEE Transactions on Signal Processing*, 62(16):4114–4128, August 2014.

[15] J. Allworth, L Windrim, J. Wardman, D. Kucharski, J. Bennett, and M. Bryson. Development of a high fidelity simulator for generalised photometric based space object classification using machine learning. In *Proceedings of 70th International Astronautical Congress (IAC)*, 2019.

[16] W. Romanishin. *An Introduction to Astronomical Photometry Using CCDs*. University of Oklahoma, 2002.

[17] Government of Canada. Radasat satellites: Technical comparison, 2021. Accessed: 2024-07-29.

[18] T. G. Stern and N. Walmsley. High-performance, reliable solar power for smallsat constellations. In *27th Annual AIAA/USU Conference on Small Satellites*, 2013.

[19] Ariane Group. Thrust and precision into space. https://www.space-propulsion.com/brochures/bipropellant-thrusters/, 2021. Accessed: 2024-07-29.

[20] H. Hakima, M. C.F. Bazzocchi, and B. Almstrom. Analysis of satellite drag coefficients based on physical and orbital specifications. In *2022 IEEE Aerospace Conference (AERO)*, pages 1–11, 2022.

[21] Space-Track. Space-track.org: Website for space object tracking data, 2024. Accessed: 2024-07-25.

[22] Orekit Development Team. Orekit: Space flight dynamics library. https://www.orekit.org/, 2024. Accessed: 2024-08-12.

[23] R. Kanzler, T. Schildknecht, T. Lips, B. Fritsche, J. Silha, and H. Krag. Space Debris Attitude Simulation - IOTA (In-Orbit Tumbling Analysis). In *Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)*, 2015.

[24] J. T. VanderPlas. Understanding the lomb–scargle periodogram. *The Astrophysical Journal Supplement Series*, 236(1):16, May 2018.

[25] T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA, 2016. Association for Computing Machinery.

[26] S. S. Dhaliwal, A. Nahid, and R. Abbas. Effective intrusion detection system using XGBoost. *Information*, 9(7), 2018.

[27] A. Ogunleye and Q. Wang. XGBoost model for chronic kidney disease diagnosis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 17(6):2131–2140, 2020.

[28] B. Naul, J. S. Bloom, F. Pérez, and S. van der Walt. A recurrent neural network for classification of unevenly sampled variable stars. *Nature Astronomy*, pages 151–155, 2018.

[29] F. Karim, S. Majumdar, H. Darabi, and S. Chen. LSTM fully convolutional networks for time series classification. *IEEE Access*, 6:1662–1669, 2018.

[30] G. P. Badura and C. R. Valenta. Physics-guided machine learning for satellite spin property estimation from light curves. In *The Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)*, 2023.