Recurrent Neural Network Autoencoders for Spin Stability Classification of Irregularly Sampled Light Curves

Gregory P. Badura, Christopher R. Valenta , Layne Churchill, Douglas A. Hope
Georgia Tech Research Institute

ABSTRACT

Providing real-time updates on satellite spin-stability is a critical task for continuously assessing collision risk. Space Domain Awareness (SDA) community analysts often manually inspect satellite light curves to assess whether a satellite is stabilized or tumbling. While human analysts can perform stability classification, the ever-growing volume of data collected and published to repositories such as the Unified Data Library (UDL) necessitates the development of explainable, automated Machine Learning (ML) stability assessment tools to assist them. Previous efforts to assess satellite stability using ML solutions assume idealized assumptions about the light curve data quality, such as having evenly time-sampled data, and full-night observations. Due to factors such as limited telescope tasking time and weather fluctuations, data uploaded to the UDL rarely meets these assumptions meaning that a significant amount of useful UDL data is under-utilized.

To address this gap, we developed an autoencoder-based anomaly detection algorithm that provides spin-stability assessments on light curves as short as 5 minutes in duration with up to a 91% score in balanced accuracy. The autoencoder was trained to reconstruct noisy light curves with explicit knowledge of the phase angle and visual magnitude uncertainty of each data point. We quantitatively and qualitatively show that the inclusion of this ancillary light curve information provides the potential for the network to overcome environmental noise, weather corruption, and uneven sensor sampling cadences in the process of learning to differentiate typical behaviors of stable light curves (i.e. glints) from those of tumbling light curves (i.e. oscillations, aliasing). The autoencoder's light curve reconstruction prediction accuracy was found to serve as an "anomaly score" for discriminating unstable light curves from stable light curves.

The autoencoder's uncertainty-weighted mean reconstruction error was combined with several expert-derived features as a feature embedding for a Balanced Random Forest (RF) classifier. The Balanced RFs that were trained using the autoencoder's reconstruction error metric outperformed those trained solely using expert level features by 9±3.5% in terms of Matthews Correlation Coefficient (MCC) and 8.5±3.7% in terms of F1-score, with a significance level of $p < 1 \times 10^{-5}$. The predictions made by the autoencoder-based system are highly interpretable to ML novices via their interpretation as an "anomaly score", meaning that they serve as a useful compliment for human analysts to understand why the system made an assesment. Our system's performance suggests that ML-derived solutions can significantly assist human analysts in real-time stability assessment from optical light curve datasets. Finally, we also present initial results on future extensions of this machine learning architecture for tasks such as light curve forecasting, continual learning, and spin periodicity estimation.

1. INTRODUCTION

The rise of global sensor networks and the increasing number of Space Objects (SOs) that are being tracked are leading to an ever-growing volume of photometric data being uploaded to Space Domain Awareness (SDA) repositories. Commercial and government providers are continuously pushing light curve observations of Geostationary (GEO) and Low Earth Orbit (LEO) satellites to Air Force Research Laboratory's Unified Data Library (UDL) from sensor sites around the globe [13]. The high cadence with which these light curves are uploaded to the UDL makes real-time analysis by human analysts unsustainable. Additionally, human analysts cannot always make use of latent information in light curve datasets (collection geometries, sampling frequencies, time-series lengths, noise characteristics, etc.) that depend on factors such as the data provider's sensing system, tasking schedules, and post-processing algorithms.

These issues necessitate the development of Machine Learning (ML) solutions for real-time and geography-independent satellite operational assessment. Recent research in the SDA community has shown promise for the ability of ML systems trained on UDL light curves to achieve this goal. In one particularly relevant study, Dao et al utilized a random

forest algorithm that ingested light curve derived features to classify GEO satellites as either tumbling or three-axis stabilized with up to 90% accuracy [10]. In another study, Haynes et al devised a statistical approach for fusing UDL light curves with varying collection geometries, seasonality, and sampling properties [17]. Numerous other studies have shown the potential of ML algorithms to classify SO properties such as attitude state, orientation, shape, and optical cross section from simulated light curve and spectroscopy datasets [3, 15, 20, 23, 25, 26, 28, 33].

Despite this initial promise of ML systems, there are many remaining practical challenges that have been unaddressed for performing classification on SO light curve data in a near real-time manner. This paper proposes an algorithm that solves several such problems, including: (1) performing classification on unevenly sampled light curves, (2) performing classification on short duration light curves, and (3) incorporating physics and measurement information to improve the relationship of the ML predictions with physical parameters. A summary of these issues in the context of our study is provided below in order to clarify the problems that they pose for light curve classification.

1. Unevenly Sampled Light Curves

"Unevenly sampled" means that a dataset is made up of light curves that do not have equal lengths or that have varying time gaps in between data points. Unevenly sampled light curves are frequently uploaded onto the UDL due to factors such as local weather complications and sensor noise. Popular ML classification algorithms such as Convolutional Neural Networks (CNNs) are typically trained using input sequences that have a fixed number of timesteps and an even sampling cadence [19]. Approaches exist for pre-processing unevenly sampled time sequences prior to CNN training, but these normally boil down to either interpolating between timesteps or padding the time series with constant values as can be demonstrated in Fig. 1 [19, 32]. While results have shown that padding light curve datasets has a limited effect on CNN performance [2], it is possible that warping or interpolation can introduce artifacts that degrade classification performance. In this paper, we developed an improved system that explicitly incorporates uneven timestep sampling into the training phase in order to avoid any padding or warping of the raw light curve.

2. Short Duration Light Curves

In the context of this study, a "short duration" light curve is defined as one having a total length that is a fraction of a full-night of observation time. The light curves that were downloaded from the UDL for usage in this paper were on frequently on the order of 10 minutes to 30 minutes in duration. Datasets of this length are pushed onto the UDL because of limitations in sensor tasking time; if a narrow-field of view telescope is tasked with observing thousands of objects over the course of a night, then each individual SO can only receive so much attention [14]. Full-night observations are easier to make classifications on because one can utilize the full span of light curve longitudinal phase angles that has been shown to possess a wealth of information on information such as solar panel pose [8] and satellite stability [10]. Short duration light curves only sample a small range of phase angle values from this span, but the high-cadence at which they are pushed to the UDL makes them a potentially valuable tool for real-time assessment of spin-stability.

3. Incorporation of Geometry and Measurement Information into Machine Learning Model

Incorporation of available physics information into a ML algorithm improves the correspondence of model predictions with physical processes. A major limitation of many supervised ML models is that they operate as a "black-box" that is oblivious to underlying processes driving a data-point's formation and measurement [11]. These models may therefore be useful for generating deceptively accurate predictions on data that is similar in physical nature to the data that it was trained on. However, the model may not generalize well on data outside of the bounds of the training dataset's parameters [34]. Additionally, incorporating physics information can in many cases improve the interpretability of the results for non-expert users. This result has the benefit of increasing user trust in the algorithm results and allowing an operator to spot an anomalous prediction more readily. In this paper, we propose an algorithm that achieves superior interpret-ability and physical correspondence to previous efforts by making use of readily available collection geometry and measurement uncertainty information.

This paper proceeeds according to the following structure. In Section 2, we provide a problem overview of performing spin stability on UDL light curves with the attributes that were just mentioned. In Section 3, we describe our machine learning autoencoder approach for handling uneven time sampling and incorporating geometry information when retrieving useful light curve features. Then in Section 4, we show examples of the performance and perform a qualitative analysis of the features that are extracted by this autoencoder architecture. In Sections 5 and 6 we provide a detailed analysis of some classifiers that can be derived from this model and detail the performance of each model. Finally,

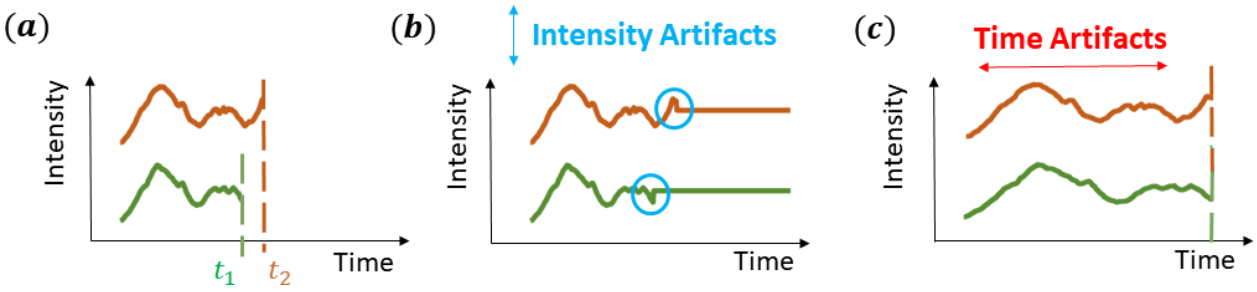


Fig. 1: Example of pre-processing steps typically employed on unevenly sampled time series: (a) two time series signals of uneven length due to different sampling durations; (b) padding the time series to even lengths, creating intensity discontinuities; (c) interpolating the data to even lengths, leading to temporal stretching.

in Section 7 we provide an overview of the potential for the model to aid in spin rate retrieval of satellites in future studies.

2. SPIN STABILITY CLASSIFICATION ON IRREGULAR DATASETS

Our study focused on classifying light curves in near real-time for spin stability as the satellite transitioned from stable operation into a tumbling mode. Therefore, we classify objects that exhibited tumbling motion as being unstable. A detailed overview of the traits that stable and tumbling satellite light curves exhibit is detailed in many other studies including [2] and [10] for the interested reader.

Satellite breakups and debris shedding events are significantly more likely to occur for satellites that are tumbling [31]. In the case of a satellite breakup, it is critical to be aware of the event quickly in order to alert nearby satellite operators of actions that can be taken to mitigate the risk of collision with debris. Making use of short-duration light curves as they stream onto the UDL as opposed to waiting for receipt of a full night of observation data can be highly useful for minimizing the time-to-response by fellow operators. An example of the utility of high-cadence light curves for alerting operators was recently demonstrated during AMC-9's loss of control in June of 2017. The light curve data was analyzed by ExoAnalytic Solutions analysts as AMC-9 transitioned from stable to unstable due to an energetic event, and the stability was documented over the full period of transition [9]. While this event demonstrates the utility of light curves for spin stability assessment, the increasing congestion of space requires that automated tools be developed to aid analysts as data volume and refresh rate exceeds human processing capabilities.

Beyond the challenge of making spin-stability assessments in a near real-time manner, an additional challenge is that many telescope operators will collect data for differing durations and at varying sampling cadences. An example of this can be seen in a dataset that our model was deployed on during a Sprint Advanced Concepts Training (SACT) Event. Our role at the SACT event focused on providing 24-hour custody of space object behaviors for a full week in December of 2021 using data that was uploaded by commercial telescope providers from around the globe [16]. Viable light curves were parsed based on the criteria that they (1) had a duration of at least 10 minutes, and (2) had a maximum gap in between any two data points of less than 30 seconds. Fig. 2 shows an example of the length of observations, number of data points, and median lag time in between successive measurements. From these distributions, it is clear that any practical high-cadence assessment tool must overcome light curve irregularity when performing spin-stability classification on datasets that are pushed onto the UDL from telescopes with differing collection paradigms.

3. AUTOENCODERS INFORMED BY OBSERVATION GEOMETRY AND MEASUREMENT ERRORS

The challenges of uneven time sampling in assessing light curve stability motivated the development of a solution that was able to capture the time-dependence of trends that were occurring in light curves. We therefore chose to rely on sequential machine learning units as opposed to hierarchical models when developing a system for extracting features from input light curve time series. This section highlights the logic behind that decision and the model that was developed using sequential architectures.

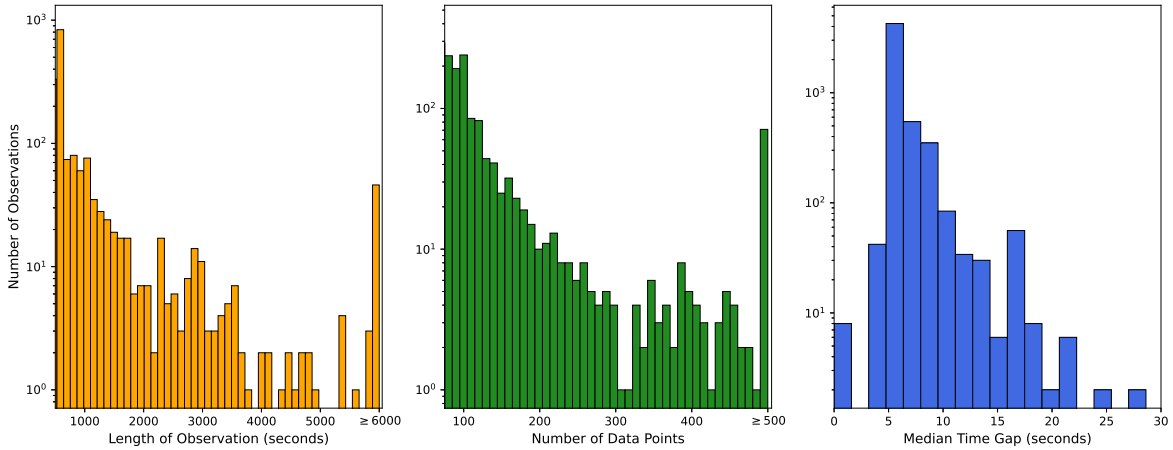


Fig. 2: Examination of the attributes of data pulled from the UDL. Histograms are shown for (left) length of observations, (middle) number of data points, and (right) median lag time in between successive data points.

3.1 Advantages of Recurrent Neural Networks over Hierarchical Models

Hierarchical machine learning models such as CNNs are trained to extract the most informative features from a fixed-length time sequence using convolutional filters [35]. These models are typically trained without a sense of the sequential nature of the time-series and instead only attempt to recognize time-invariant features in the signal [19]. In order to accomplish this objective, hierarchical techniques for light curve analysis typically rely on assumptions that the light curve is uniformly sampled in time and that the lengths of all light curves in the training dataset are equal [3]. One of the most common techniques for feeding a new time-series that does not meet a trained CNN model's original sampling assumptions is interpolation [32, 19]. A major flaw of interpolation is that it assumes that a regression can account for the behaviors in between successive data points. This regression can introduce biases and artifacts that grow proportionally to the time gaps in between individual brightness measurements [27].

Recurrent Neural Networks (RNNs) are advantageous for handling irregular time sequences because they are trained to treat input time-series in a sequential manner [12]. In other words, RNNs retain information from one time step to the next through the incorporation of a hidden state. This hidden state allows an RNN unit to model trends in the input signal as a function of time by considering both patterns that occurred over the previous timesteps, as well as the input information at the current timestep [35]. The two dominant archetypes of RNNs currently deployed in the machine learning field also employ "gates" that allow the RNN units to either forget or propagate hidden state information that is most relevant to the training task: (1) the Long Short Term Memory (LSTM) [18], and (2) Gated Recurrent Unit [6]. Both of these models were considered in the hypertuning of the autoencoder model of this study for their ability to learn time-dependent trends in the data that are most important for reconstructing stable and unstable light curves.

These two models are covered exhaustively in the literature (e.g. [35, 36]). Therefore, we consider the structure of the best performing RNN, the LSTM, in order to demonstrate how the use of "gates" can be beneficial for sequential modeling of time series. The core component of an LSTM is the memory cell that uses a hidden state (h_t) and a cell state (c_t) to capture temporal trends in a time series. The LSTM controls these states using three different gate mechanisms: (1) an input gate (i_t) that determines if input visual magnitude data at the current timestep (m_t) is relevant to recent trend information, (2) a forget gate (f_t) that decides if the previous internal state (h_{t-1}) is relevant to any ongoing light curve trends, and (3) an output gate (o_t) that decides the next hidden state (h_t) based on consideration of previous output information and internal cell state.

Mathematically, the LSTM for a light curve time sequence can be described by the following equations [36]:

$$\begin{pmatrix} f_t \\ i_t \\ o_t \\ g_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} \mathbf{W} \begin{pmatrix} h_{t-1} \\ m_t \end{pmatrix} \tag{1}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{2}$$

$$h_t = o_t \odot \tanh(c_t) \tag{3}$$

where the sigmoid (σ) function and tanh function are applied element-wise, g_t is used to modulate the input gate's contribution to the the cell state c_t, $\mathbf{W}$ is a weight matrix that is trained via backpropagation, and $\odot$ signifies the Hadamard product.

Equation 1 demonstrates how the three gates provide the ability to control the cell state across timesteps. The sigmoid layer by definition will return a value between zero and one, providing a simple way to control which information is let through the input, forget, and output gates. Essentially, a value of 0 means that information is blocked from propagation to the next timestep, while a value of 1 means that information will contribute to the next timestep [18]. Equations 2 and 3 are then where the LSTM combines information that streams from all gates in order to modify both the cell state and hidden state. By controlling which information is allowed to pass from timestep $t-1$ to timestep t, the model is able to allow relevant time-dependent (or in the case of this paper, geometry-dependent) past trends to persist in the memory of the cell as input data streams into the architecture. In summary, the combined effect of the LSTM's gates and internal states enable it to model trends in the light curve series in a sequential and time-variant manner. These traits are highly advantageous for handling irregularly sampled light curves, as we will show in the next sub-section.

The GRU model takes on a very similar form to the LSTM with the exception that the model does not have a memory unit, and instead only relies on a hidden state [6]. The equations of the GRU model take on the following form [35]:

$$z_t = \sigma(m_t \mathbf{W_z} + h_{t-1} \mathbf{U_z}) \tag{4}$$

$$r_t = \sigma(m_t \mathbf{W_r} + h_{t-1} \mathbf{U_r}) \tag{5}$$

$$s_t = \tanh(m_t \mathbf{W_s} + (h_{t-1} \odot r_t) \mathbf{U_s}) \tag{6}$$

$$h_t = z_t \odot s_t + (1 - z_t) \odot h_{t-1} \tag{7}$$

where r is the reset gate, z is the update gate of the GRU, s is the candidate activation gate, and ($\mathbf{W_k}$, $\mathbf{U_k}$) are trainable parameter matrices for the k^{th} gate [35]. As can be seen. the general structure of the LSTM (Equations 1 through 3) and the GRU (Equations 4 through 7) are very similar in the use of activation functions and number of gates.

Studies have founds that the GRU model and the LSTM model generate comparable performance on many sequential machine learning tasks [7]. In practice, tuning hyperparameters like layer size have been shown to be more important than picking the ideal architecture [35]. Therefore, in our study we considered both hypertuning and RNN architecture when choosing the optimal model for the task of light curve reconstruction on irregularly sampled light curves.

3.2 Autoencoder Model Employing Recurrent Neural Networks

In order to overcome uneven temporal sampling issues encountered in light curve analysis of variable stars, Naul et al proposed a novel RNN-based autoencoder algorithm specifically made for time-series analysis [27]. The encoder and decoder components of a standard autoencoder work in tandem to generate a feature embedding from an input time series, and then reconstruct the original time series using the learned feature embedding. The principal advantages of

Naul et al's sequential autoencoder over a standard autoencoder were (1) the explicit incorporation of sampling times into the time-distributed layers and (2) the explicit use of measurement uncertainty in the loss function [27].

This algorithm presents natural extensions to light curve analysis of geostationary space objects due to the time series sampling issues that were outlined in Section 2. We therefore adapted the algorithm for the purposes of SDA light curve analysis using the framework outlined in Fig. 3. There were two major changes that were made in order to adapt the algorithm for space object light curve analysis in our paper: (1) phase angle was used in place of time step information in order to allow the system to learn phase-angle dependent trends that have been shown to be relevant to learning solar-observer geometry influenced trends in stable satellites [8, 21], and (2) visual magnitude uncertainties from the UDL were used in order to minimize the contribution of noisy data points in obtaining an optimal light curve reconstruction.

In the encoder portion of our autoencoder, the observed light curve data points (m_i) and corresponding phase angle of each point (ϕ_i) are fed into the RNN layers of the encoder as a $(2 \times n)$ matrix, where n is the discrete number of data points in the light curve. The RNN units that make up each of the RNN layers maintain hidden states that can propagate important trends across units, as indicated by the red arrows in Fig. 3. The hidden RNN layers are therefore able to learn how to model short term dependencies that occur in light curves (i.e. glints and maneuvers) as a function of the phase angles at which these phenomena are most likely to occur. The final RNN layer of the encoder model produces a fixed-length embedding of shape $(m \times n)$, where m is equal to the number of hidden units in the RNN layer. This embedding has m extracted features that are essentially the most important components necessary for reconstructing the input light curve. In other words, this embedding can be thought of as a feature extraction on the input visual magnitude and phase angle measurements [27].

This embedding is then passed to the decoder component of the autoencoder, whose major task is to translate the feature embedding representation back into the original light curve. The first step that occurs in our network is that the phase angle measurements are concatenated to the feature embedding, producing a $(m+1 \times n)$ matrix. This data is then fed into a second set of hidden RNN layers that are independently trained relative to the encoder's hidden RNN layers. These decoder hidden layers learn how to combine short term dependencies across the time-dimension in order to produce a light curve reconstruction $\vec{m}^*$ that is of the same shape $(1 \times n)$ of the input visual magnitude data.

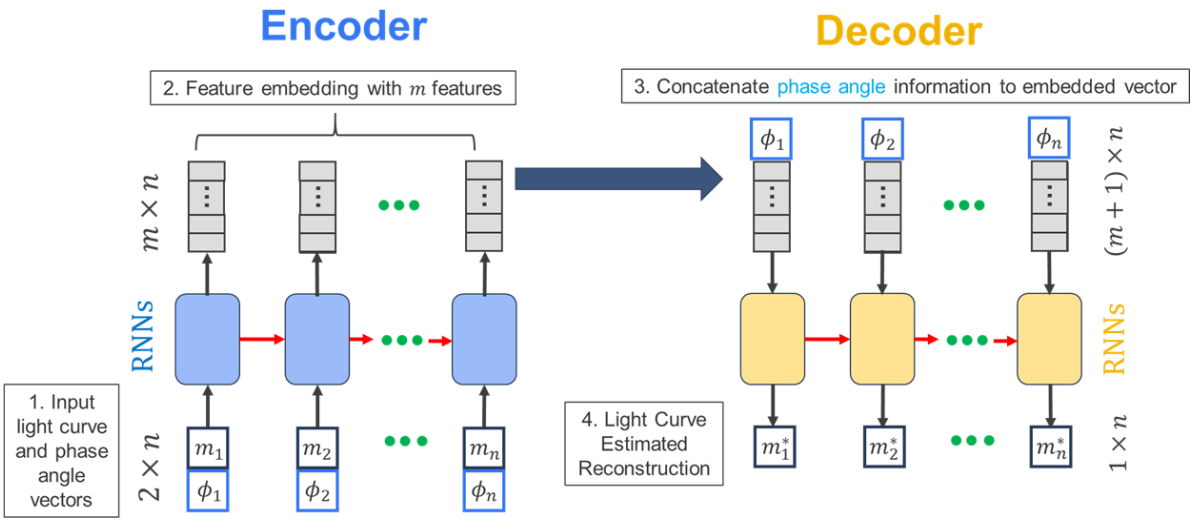


Fig. 3: A simplified model of the autoencoder for uneven time series. The encoder (left) accepts visual magnitude values and phase angle information as inputs. The input data time series is processed by a hidden recurrent layer to produce a sequence of extracted features (Step 1). A fixed-length embedding that uses a linear activation function accepts the output of the final hidden recurrent layer to produce the final embedding (Step 2). The decoder (right) accepts this embedding as input, and then concatenates the phase angle information to it (Step 3). The decoder then translates the feature embedding representation into a light curve reconstruction via its own sequence of hidden recurrent layers (Step 4).

The autoencoder network outlined in Fig. 3 is trained with phase angle, visual magnitude measurements, and visual

magnitude uncertainties as the input data and those same visual magnitude measurements as output data. The mean squared reconstruction error of the predicted visual magnitude values, m_i^*, are minimized using provider uploaded visual magnitude uncertainties, σ_i, as weights in order to reduce the contribution of noisy data points to the reconstruction error being minimized through the process of backpropagation and gradient descent. This is a key component for enabling our autoencoder to not only overcome irregularly sampled data, but also to overcome poor quality data that is typically encountered on bad weather nights. The weighted mean squared error (MSE) that is minimized in our training process can be written as:

$$MSE = \sum_{j}^{K} \sum_{i}^{N_j} \frac{(m_{j,i} - m_{j,i}^*)^2}{{\sigma_{j,i}}^2} \tag{8}$$

where K is the total number of light curves used for training, and N_j is the total number of data points in the j^{th} light curve.

In many cases the autoencoder's principal task is to learn a high quality feature extraction that can be used to build a model to solve the supervised classification task [27]. Therefore, the decoder is normally discarded and the encoder's trained weights are used as input to fully-connected layers that make use of the feature embedding for the new task. However, recent studies have shown that the autoencoder reconstruction error itself can be used as a feature to a classifier for the purposes of anomaly detection [1]. As we will show in our Results section, our trained autoencoder's reconstructions can be seen to shed light on whether a satellite is stabilized or unstabilized due to the lack of a periodic function within the weighted MSE of Equation 8. Therefore, we considered two different classifiers in this study that will be discussed in the proceeding sections: one that discards the decoder, and one that keeps the autoencoder intact.

4. AUTOENCODER TRAINING

In this section, we outline the processes that were used in order to hypertune our autoencoder model and the observations that were made regarding the optimally tuned model's performance. Optimally training the autoencoder on our dataset was the first step towards deriving a classifier that directly relied on the embedded features of the autoencoder model.

4.1 Dataset

As was mentioned in Section 2, the light curves that are pushed onto the UDL have varying temporal properties in terms of duration for and time lags. In order to filter out unusable measurements, we set three major criteria for incorporating a light curve into our dataset: the light curves must have a maximum time lag in between any two successive data points of 30 seconds, the light curves must be at least of 50 data points in length, and the light curve must last for at least 500 seconds in duration. These criterion was chosen to mitigate issues with classifying light curves that were of insufficient length for recognizing oscillations in the light curve that are caused by satellite tumbling.

A python script was written to pull down UDL light curves over a period of approximately 2 years prior to December 2021. These light curves were pulled entirely from one commercial data provider: ExoAnalytic Solutions. ExoAnalytic Solutions has ground-based sensors spread across the globe that provided regular revisits of the satellites of interest to this study across the entire year. Their datasets also provided the necessary ancillary information of phase angle and measurement uncertainty that was critical to successful autoencoder reconstructions.

In total, approximately 4800 light curves were downloaded from the UDL that met our stipulations, with approximately 40% of them being categorized as unstable. The labeling of the dataset was done with assistance from the SACT team in preparation for algorithm deployment at the December 2021 SACT event [16].

4.2 Hypertuning and Optimal Model

In order to hypertune the autoencoder model that is illustrated in Figure 3, we trained a total of 180 different models using parameters that were informed by the original study [27]. Table 1 shows these parameters that were tuned, and the optimal model that was chosen based on having the lowest overall weighted MSE on a validation dataset comprising 20% of the original dataset. For each model trained, a patience value of 50 epochs was used to monitor the validation loss and ensure that overfitting did not occur.

Table 1: Parameters that were varied in the hyperparameter tuning of the autoencoder model and the optimal model parameters derived as a function of the training process.

Hypertuning Parameter	Tuning Options	Optimal Tuning Option
RNN Type	GRU, LSTM	LSTM
Number of RNN Hidden Layers	1, 2, 3	2
RNN Hidden Layer Units	16, 32, 64	64
Feature Embedding Size	8, 16	16
Learning Rate of Adam Optimizer	1E-3, 1E-4, 1E-5, 1E-6, 1E-7	1E-04

The "Number of RNN hidden Layers" denotes the total number of layers that preceded the linear activation embedding layer. These are layers in which the hidden states were shared amongst the RNN units of either type GRU or LSTM, depending on the model being trained. The "Hidden Layer Units" denotes the total number of features in each of these hidden layers, and was assumed to be constant across all hidden layers currently being trained. Note that per [27]'s recommendation, we take the number and size of recurrent layers in the encoder and decoder modules to be equal despite the fact that they are independent units that do not share any weights. Also note that a dropout rate of 20% was used after each of the RNN hidden layers. Finally, the "Feature Embedding Size" denotes the total number of features in the linear dense activation layer that accepted input from the final RNN hidden layer.

4.3 Autoencoder Light Curve Reconstruction Examples

In order to understand the phase-angle dependent trends that were being captured by the autoencoder, we monitored the output features of the encoder's embedding layer and the decoder's reconstruction of the light curve. This information was useful for understanding the differences in extracted patterns between the stable light curve class and the unstable light curve class.

An example of the encoder's and decoder's output data products are shown in Fig. 4 for a stable light curve from our validation dataset. This stable light curve showed the expected characteristics of being smoothly varying as a function of phase angle [2, 10]. The bottom figure shows the raw visual magnitude data in blue and the autoencoder's reconstruction of the data in red. The visual magnitude uncertainties are also shown via error bars. The top figure in this image shows the 16 embedded features that were extracted by the hidden RNN layer.

It can be seen that the input stable light curve contained two noisy data points with high visual magnitude uncertainty in the regions of $\phi = 16.8^\circ$ and $\phi = 17.9^\circ$. From a human analyst perspective, these noisy points could potentially be confused with glints or maneuvers occurring in the time series, and it is therefore of interest to see how the autoencoder treats these data points. These noisy measurements are strongly responded to by the encoder's embedding layers, with responses in the region of $\phi = 16.8^\circ$ almost seeming to resemble different wavelet family responses. The embedding layers seem to be trained to respond with unique sign changes depending on the deviation of the data point from the trendline of visual magnitude as a function of phase angle. For example, embedding layer #13 responds negatively to the deviation at $\phi = 16.8^\circ$ while the embedding layer #5 responds positively to the same deviation. Despite these strong embedding feature responses, the autoencoder learns through the incorporation of visual magnitude uncertainty in the weighted MSE that these data points should not strongly affect the autoencoder reconstruction, as is shown via the red trendline in the bottom plot of Fig. 4.

An example of our autoencoder's output data products are shown in Fig. 5 for a tumbling light curve from the validation dataset. This light curve is aliased likely because telescope's sampling frequency was below the Nyquist rate of the spinning of the satellite. Despite this aliasing, the embedding layers seem capable of recognizing the tumbling that is occurring within the signal in the form of repeatable periodic patterns that are present in the encoder embedding vectors (i.e. embedding #1, #4, and #5 in top image of Fig. 5). Despite the apparent periodicity in the feature embeddings, the output autoencoder reconstruction does not have any significant periodicity in its structure. The light curve reconstruction is quite poor and seems to hug the lower rung of visual magnitude values, resulting om a bi-modal distribution of reconstruction errors that is large for the top rung of aliased points and small for the bottom rung of aliased data points. In summary, the autoencoder did not perform reconstructions well on aliased signals, suggesting that a feature representation should be developed to flag aliased light curves.

An example of our autoencoder's output embeddings and reconstructions are shown in Fig. 6 for a tumbling light curve

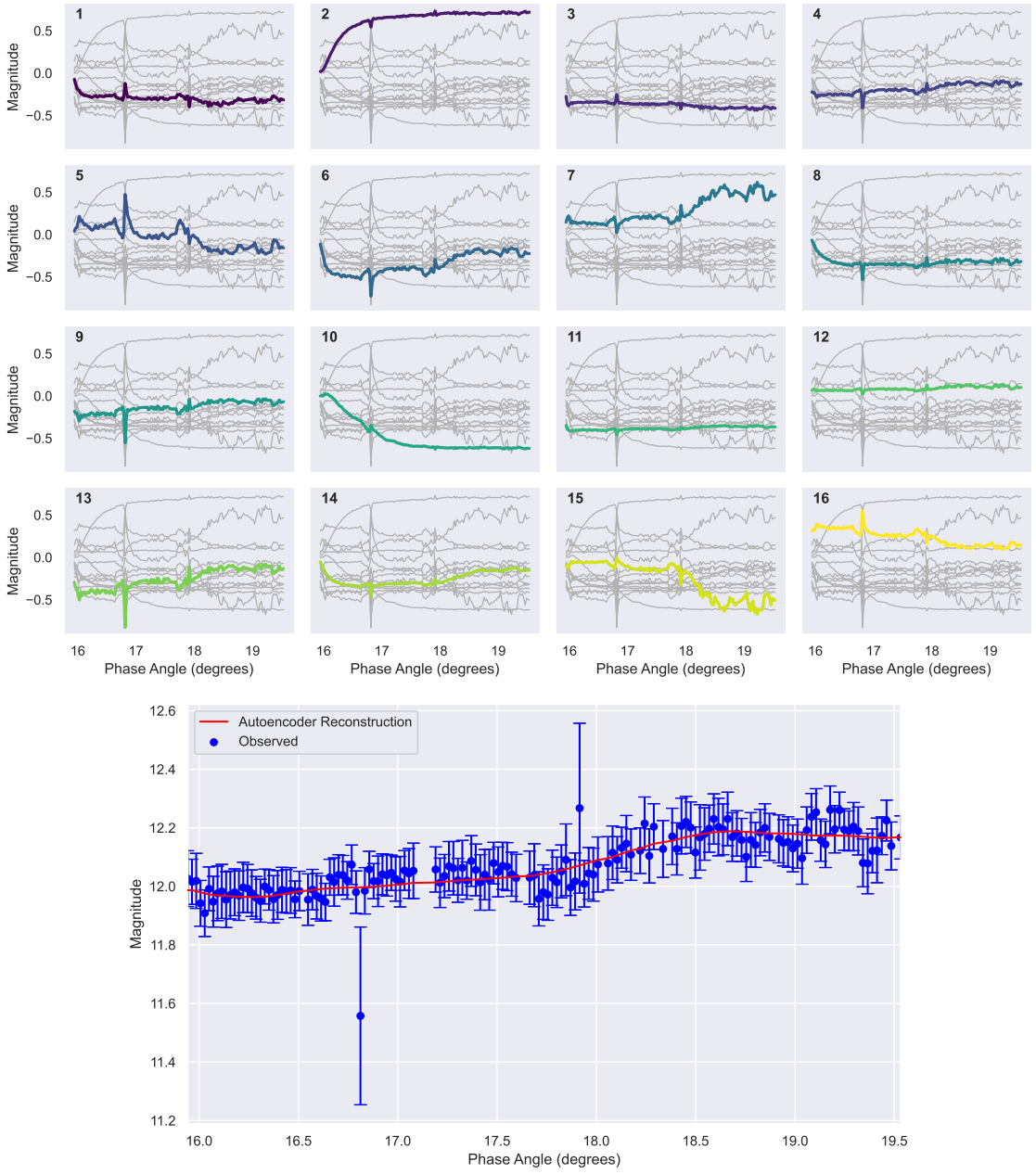


Fig. 4: The optimally trained autoencoder's light curve reconstructions for an example of a stable satellite. (Top) Responses of embedding layers to the stable light curve. (Bottom) The raw data input to the autoencoder (blue) and the output reconstruction of the decoder (red).

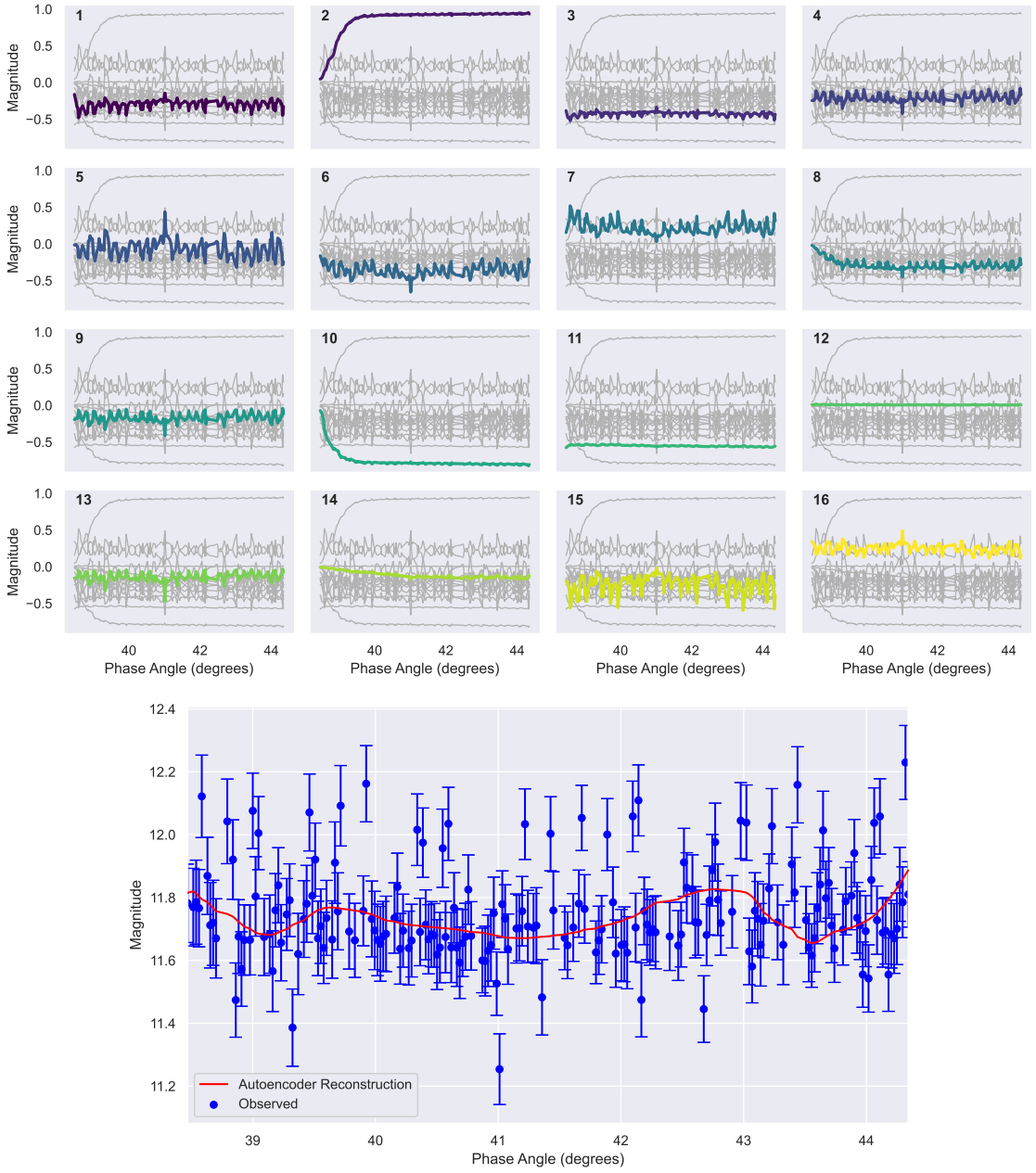


Fig. 5: The optimally trained autoencoder's light curve reconstructions for an example of an aliased signal of an unstable satellite light curve. (Top) Responses of unique embedding layers to the light curve. (Bottom) The raw data input to the autoencoder (blue) and the output reconstruction of the decoder (red).

that was not aliased (i.e. it was properly temporally sampled). We can see that the embedding layers seem capable of capturing the tumbling sequence that occurs in the input light curve. It also appears that certain embedded feature oscillations such as those from embeddings #5 and #7 match the true periodicity of the light curve. This result suggests that the encoder's embedding layers learned from this study can potentially be transferred to different tasks such as synodic spin rate estimation in future studies. Despite the apparent periodic signals that are extracted by the embedded vectors, periodicity is not strongly modeled in the reconstruction. This results in an autoencoder reconstructed light curve does not match the input light curve's peaks and troughs. This behavior was noted by Naul et al in their study where they found that unless the input time information was period-folded, that the autoencoder would struggle to

reconstruct a variable light curve properly [27]. However, period-folding requires a prior-estimate of the satellite's spin rate, which was beyond the scope of this study. This application will be extended in a future extension of the current model architecture.

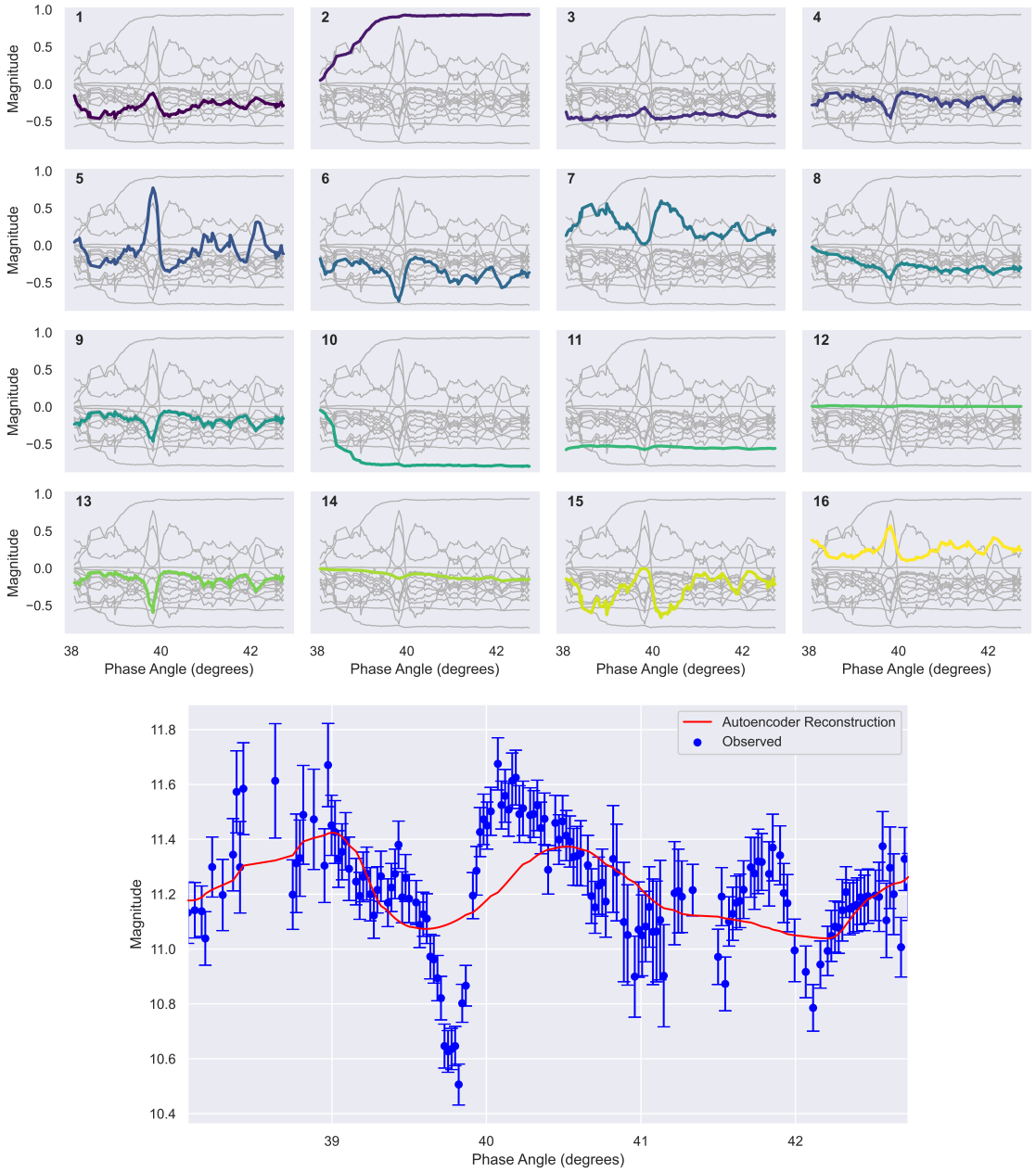


Fig. 6: The trained autoencoder's reconstructions for a properly sampled unstable satellite light curve. (Top) Responses of unique embedding layers to the light curve. (Bottom) The raw data input to the autoencoder (blue) and the output reconstruction of the decoder (red).

We observed a general trend that the autoencoder's light curve reconstruction MSE was lower for stable light curves than for unstable light curves. In order to highlight this trend, we show a 2D scatter plot of the MSE vs. the light curve's mean visual magnitude for both classes in Fig. 7. Despite this general overall trend of lower MSE for stable light curves, it can be seen that there is still significant overlap between the stable and unstable classes that is marginally improved via the incorporation of mean visual magnitude as a second feature. This result suggested that incorporating

the autoencoder's reconstruction MSE as an additional discrete feature could potentially improve stability assessment classifier performance. These observations about the potential strengths and weaknesses of the autoencoder as an anomaly detector informed the development of one of the classifier models that we will introduce in the next section.

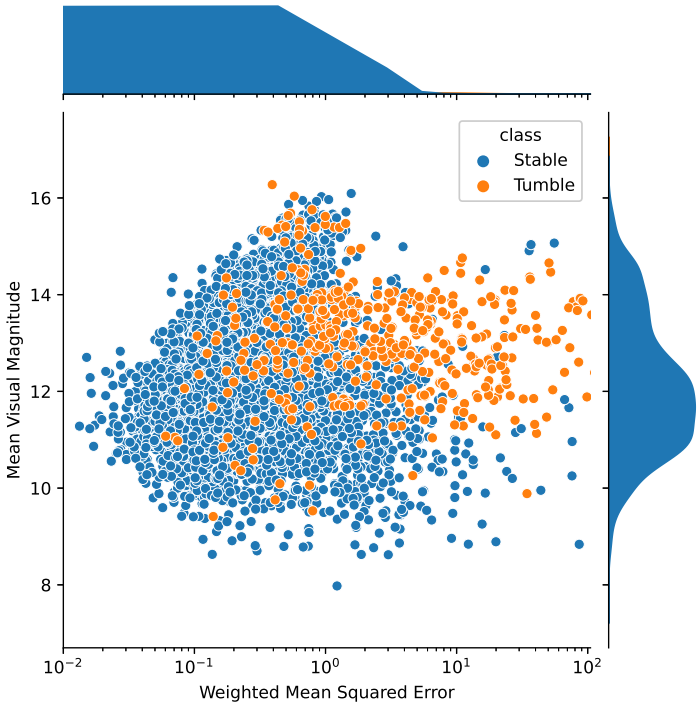


Fig. 7: A plot of the weight mean squared error vs. the mean visual magnitude for the stable and unstable light curves in this study.

5. SPIN STABILITY CLASSIFIERS

Three different spin stability classifiers were developed in this study. Two of the models were derived using the optimally trained autoencoder light curve reconstruction model that was described in the previous section. One of these models relied on the use of the autencoder's weighted reconstruction MSE as a discrete feature, while the other relied explicity on the embedded features of the encoder portion of the autoencoder model. The third model was a baseline model using SDA expert derived features. This baseline model was meant to serve as the "human analyst" performance baseline against which our autoencoder models could be compared.

5.1 Balanced Random Forest from Expert Features

Our baseline model comes from a recent study by Dao et al in which they developed a suite of statistical metrics for categorizing satellites as either stable or tumbling [10]. Dao's statistical metrics were developed based on a thorough analysis of traits that a human analyst would perceive as useful for differentiating a stable satellite from a tumbling satellite. These metrics were derived solely from visual magnitude and phase angles. Many of these metrics also made no assumptions about the irregularity of the light curve sampling. These two traits made their model an ideal comparison for our autoencoder's classification models.

All of the statistical metrics used in our baseline model and their respective descriptions are shown in Table 4 of the Appendix section. For the sake of brevity, a detailed explanation is not provided, but the interested reader can see the original paper [10]. The metrics that were used in our baseline model range from simple correlations to more complex statistical metrics. As an example, one of the metrics is a simple linear correlation of magnitude vs. phase angle that is meant to discern predictable trends that are characteristic of stable satellites. More complex metrics such as kurtosis and skewness are meant to find bi-modal distributions of magnitude values in a signal that are characteristic of aliased

tumbling signatures in which the middle rung of magnitude values are not captured [10]. Note that Dao's study, they considered the model's performance on full-night light curves. Because our study focused on short-duration light curves, we did not make use of statistical metrics from their study that relied on having the full range of Longitudinal Phase Angle (LPA) values.

Dao's model was developed using a Random Forest (RF) supervised ML algorithm, which creates a forest of decision trees and then makes a classification based on majority voting of their individual decisions [4]. We computed their metrics for each of our UDL light curves and then split the dataset into training and validation datasets using an 80/20 balance. We then optimally hypertuned a Balanced Random Forest Classifier (BRF) by evaluating model performance on the validation portion of our dataset across all model parameter combinations [22]. This hypertuning procedure consisted of varying the number of decision trees, the split criterion, the percentage of features to consider at each split, and the maximum number of samples to train each baseline estimator. The optimally tuned model resulted in the following respective values: 150, entropy, 0.33, 0.33. Note that the BRF model has the advantage of taking class imbalance into account, while a standard RF model ignores class imbalance [22]. This was advantageous given the higher percentage of stable light curves in our dataset, and we found that the BRF model provided a 20% improvement in balanced accuracy over a standard RF model.

5.2 Encoder-to-Dense Classifer

Our first model was based on making sole use of an ML model's learned sequential feature extraction. This model does not consider any human analyst derived features. In this approach, the autoencoder is trained using the full database of light curves and methods that were outlined in Section 4s. The optimally tuned model's encoder weights are then extracted from the autoencoder model and the decoder portion of the model is discarded. The extracted encoder weights are used to generate features to solve the spin-stability classification problem. The idea behind this classifier model is that the encoder portion of the model learned useful time-variant feature extraction techniques that can be transferred to other tasks besides besides light curve reconstruction. As was seen in the embedding features generated for unstable light curves in Figures 5 and 6, these time variant embedding layers appear to be able to recognize periodic trends in the input light curves that can potentially translate to accurate tumbling classification.

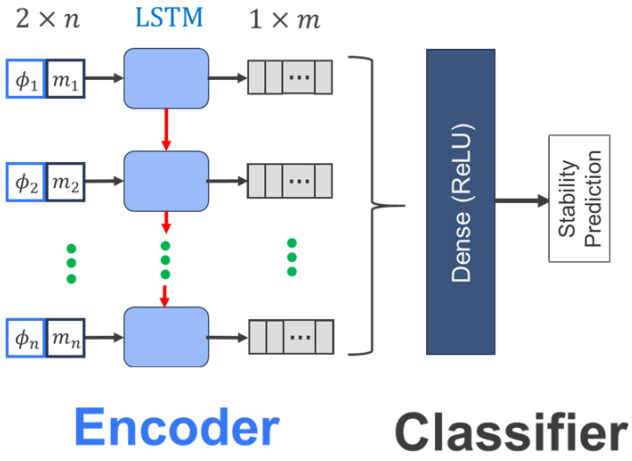


Fig. 8: The Encoder-to-Dense classifier model that was used in this study.

This model was trained by freezing the encoder parameters and weights from the optimally tuned autoencoder model that was outlined in Table 1. A hyperparameter tuning study was then conducted in order to determine the optimal arrangement of dense layers that would ingest the encoder extracted features from the encoder to perform spin-stability classification according to the architecture shown in Figure 8. The hyperparameter settings included the total number of Dense layers with Rectified Linear Unit (ReLU) activation (options: 1, 2, and 3), the dimensionality of the each of these dense layers (options: 32, 64, and 128), and the training rate of the Adam optimizer model. The hyperparameter study's optimal model had a single hidden dense ReLU layer of dimensionality 64, but all models achieved very similar performance on the validation dataset suggesting an upper bound on the ability to perform classification using the encoder's learned embeddings.

5.3 Enhanced Balanced Random Forest Anomaly Classifer

As was discussed in Section 4.3, the derived autoencoder model classifier generally performed better at reconstructing stable light curves than unstable light curves. The poor performance of the autoencoder in reconstructing unstable light curves likely resulted from (1) the cost function for weighted MSE in Equation 8 not incorporating a periodic function and (2) the lack of period-folding of the input phase angle data [27]. While fixing these issues in future studies can potentially improve the model's ability to recover spin-rate, the cost function in this study lends itself towards recognition of tumbling behaviors by treating the MSE as a type of "anomaly score." This choice is inspired by previous studies on time series having shown that a properly trained autoencoder is able to differentiate an anomalous signal from a non-anomalous signal based solely on the retrieved reconstruction MSE [1, 24].

When analyzing the correlations between the weighted MSE of our autoencoder model and the expert features from Table 4, it became evident that the autoencoder provided unique information. A diagram of this correlation plot for our 4800 light curves is shown in Fig. 9. We initially expected that the weighted MSE would be strongly correlated with the skewness and kurtosis metrics that are meant to discern bi-modality of aliased signals. However, the weighted MSE only demonstrated a weak correlation of approximately 0.4 with the kurtosis metric. Furthermore, the weighted MSE metric was only moderately negatively correlated with the R^2 metrics that are meant to aid with the task of stable satellite recognition. Interestingly, the weighted MSE was most highly correlated with the visual magnitude range metric that computes the difference of the highest and lowest visual magnitude within a light curve. This is possibly due to the range metric being correlated with the peaks and troughs of a tumbling signal, but it is unclear if this would hold over all datasets.

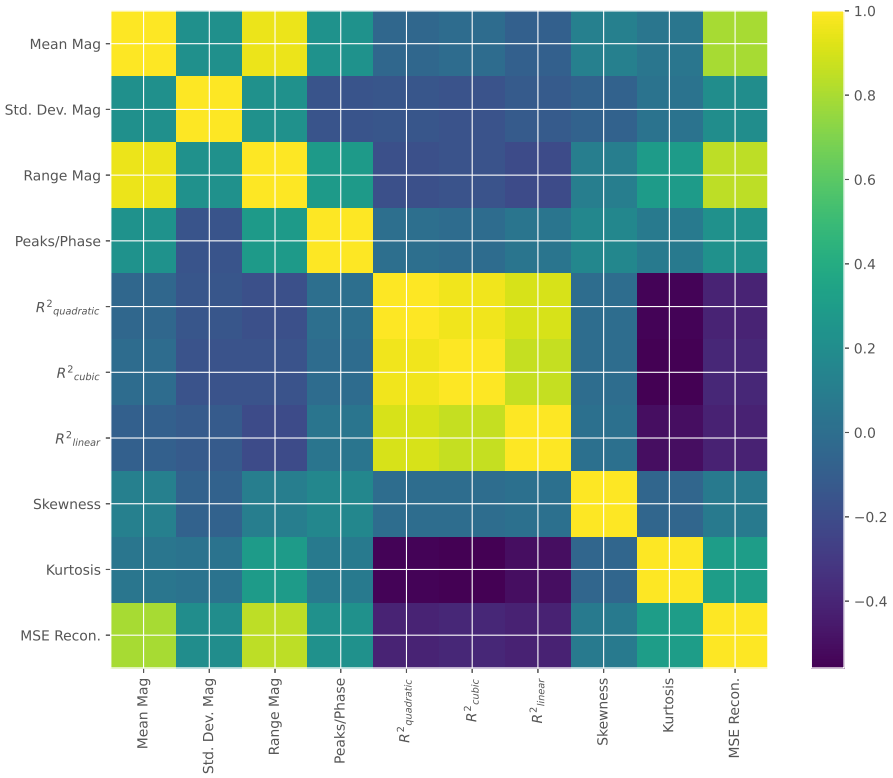


Fig. 9: A correlation matrix describing the correlation between the expert features from [10]'s study and the weighted mean squared error of reconstruction by the autoencoder from Equation 8. Table 4 provides definitions of the features labeled here.

Based on these observations, we sought to determine if the incorporation of the weighted MSE metric into a BRF classifier could enhance classifier performance relative to the baseline model. We therefore trained a model according to the architecture in Fig. 10, where the 9 expert features and the autoencoder reconstruction weighted MSE serve as the feature set. The BRF model that ingested these 10 features was hypertuned according to the same procedure

outlined in Section 5.1. Throughout the hypertuning procedure, we varied the number of decision trees, the split criterion, the percentage of features to consider at each split, and the maximum number of samples to train each baseline estimator. The optimally tuned model resulted in the same respective values as the baseline model: 150, entropy, 0.33, 0.33.

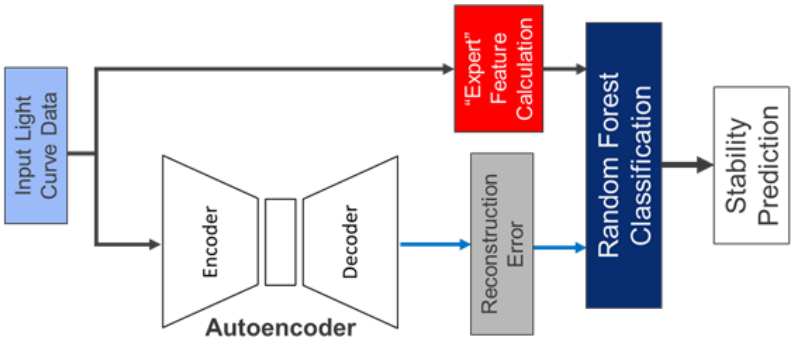


Fig. 10: The Enhanced Balanced Random Forest classifier model that relies on expert features and the autoencoder's reconstruction error.

6. SPIN STABILITY CLASSIFIER MODELS PERFORMANCE

6.1 Confusion Matrices of Architectures

In order to evaluate each architecture's classification performance, we computed the average confusion matrix of each architecture resulting from a k-fold cross validation procedure where $k = 10$. This k-fold cross validation estimates the skill of a model on unseen data by assessing the performance of an architecture in making predictions on data that was not deployed during the training of the model [29]. In order to run this cross-validation, the dataset of 4800 light curves was shuffled randomly and then split into 10 evenly split groups of an approximate size of 480 light curves. Each of these groups was kept aside as a test dataset for evaluating one of k different models. Each of these k test datasets was held aside and the remaining portion of light curves was utilized as the training dataset for the k^{th} model. After fully training the k^{th} model, the evaluation score was recorded on the test dataset and the model was discarded.

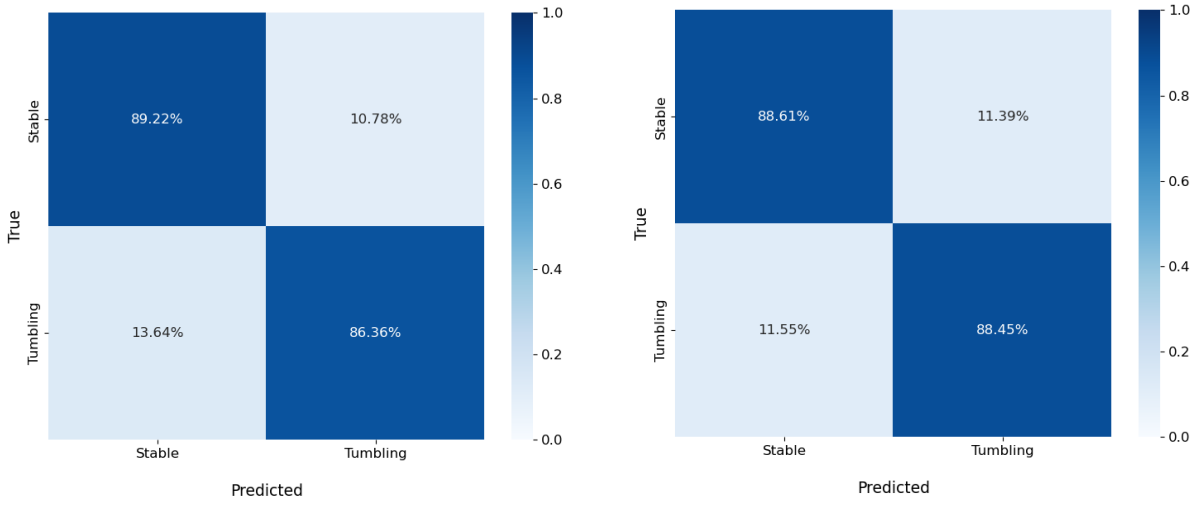
The average confusion matrix across these $k = 10$ runs for each of the architectures is shown in Fig. 11. Note that the autoencoder and encoder weights were considered fully trained and were kept frozen throughout this k-fold cross validation approach. In short, only the dense model weights and the random forest components of the models were varied in the process of training these models.

We can see that the baseline model (Fig. 11 (a)) was outperformed by the "Encoder-to-Dense" model (Fig. 11 (b)) in the task of classifying tumbling light curves by approximately 2%. Recall that the "Encoder-to-Dense" model solely relied on the autoencoder's learned encoder embeddings as its feature set. In other words, this "Encoder-to-Dense model" relied only on machine learning generated features for classification, while the baseline model relied on expert-derived features. This suggests that the encoder model's ability to extract periodic patterns that was outlined in Section 4.3 was more useful for the unstable classification task than the expert-derived features of skewness and kurtosis. On the other hand, the baseline model outperformed the "Encoder-to-Dense" model by approximately 1% in the task of accurately classifying stable satellites suggesting a limitation of these encoder embeddings for recognizing stable behaviors.

We also see that the "Enhanced Balanced Random Forest Anomaly Classifier" (Fig. 11 (c)) outperforms both the baseline model and the "Encoder-to-Dense model" in accurately assessing both stable and unstable satellites. It achieves true positive rates of approximately 91% and 90% in assessing stable and unstable satellites, respectively. This outcome suggests that merging the human-derive feature set's ability to discern stable satellites with the autoencoder's ability to recognize tumbling behaviors results in a superior set of features to either feature subset.

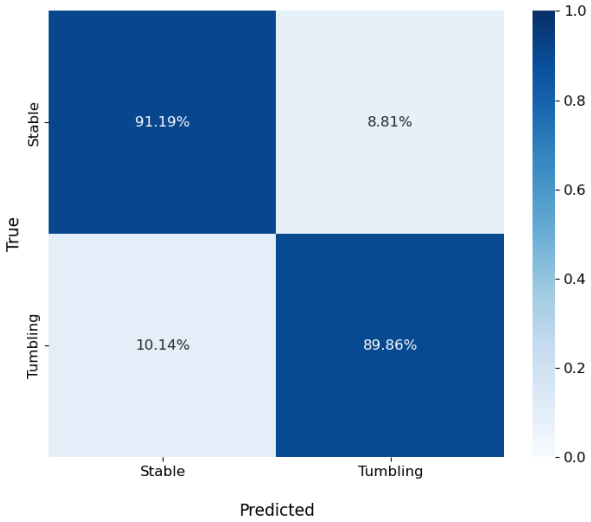
6.2 K-fold Cross Validation Comparison with Baseline Model

In order to compare each architecture's performance using standard machine learning metrics, we compute the mean and standard deviation for three different metrics across the 10-fold cross validation: F1 score, Matthews Correlation



(a)

(b)



(c)

Fig. 11: Confusion matrices showing average classifier performance across 10-fold cross-validation runs for the models in this study: (a) Balanced Random Forest from Expert Features, (b) Encoder-to-Dense Classifier, and (c) Enhanced Balanced Random Forest Anomaly Classifier.

Table 2: Average and standard deviations of model performances throughout the 10-fold cross validation procedure.

Model	F1 Score	MCC Score	Balanced Accuracy
Balanced Random Forest from Expert Features	0.601 $\pm$0.023	0.584 $\pm$0.025	88.5 $\pm$1.3
Encoder-to-Dense Classifier	0.59 $\pm$0.023	0.584 $\pm$0.023	88.3 $\pm$1.3
Enhanced Balanced Random Forest Anomaly Classifier	0.652 $\pm$0.024	0.636 $\pm$0.026	90.4 $\pm$1.7

Table 3: The results of a right-sided t-test on the cross-validation runs when evaluating null hypothesis that the baseline model outperformed the autoencoder-based models.

Model	MCC Score	P-value	Reject Null?
Encoder-to-Dense Classifier	0.584 $\pm$0.023	0.11	False
Enhanced Balanced Random Forest Anomaly Classifier	0.652 $\pm$0.024	1×10^{-6}	True

Coefficient (MCC), and balanced accuracy. These metrics are defined in Appendix Section B for the interested reader. The computed mean and standard deviation of each metric for each of the three models are shown in Table 2. From this table, we can see that the "Enhanced Balanced Random Forest Anomaly Classifier" model outperformed the baseline model by 9±3.5% in terms of MCC and 8.5±3.7% in terms of F1-score. The "Enhanced Balanced Random Forest Anomaly Classifier" model also outperformed the "Encoder-to-Dense" model by approximately the same percentage change. This result suggests that our model that combines human-expert features with autoencoder reconstruction metrics outperforms the models that are either (1) solely influenced by expert insights or (2) solely informed by machine learning insights.

While these results are promising, we wanted to verify that they were repeatable across all runs due to the relatively small amount of light curves that were available for this initial study. We therefore performed a right-tailed significance t-test on the 10-fold cross validation runs in order to test the null hypothesis $H0$ that the baseline model had greater than or equal performance to the encoder-based models [30]. The alternate hypothesis $H1$ for this t-test was that the autoencoder-based model outperformed the baseline model. The primary metric for evaluation in this t-test was the MCC metric. This choice was because MCC (1) is more reliable than $F1$, Balanced Accuracy, and precision when applied to unbalanced datasets, and (2) MCC only provides high scores when the model performs well in properly classifying all four quadrants of the confusion matrix [5]. The results of performing this t-test are on the MCC metric are shown in Table 3. In this table, we can see that we were not able to reject the null hypothesis for the the "Encoder-to-Dense" model, but that we were able to reject the null hypothesis for the "Enhanced Balanced Random Forest Anomaly Classifier" model to within a significance level of $p < 1 \times 10^{-5}$.

6.3 Model Performance As Function of Light Curve Length

We observed that all three of the classifiers in general performed better when being fed light curves that were longer in duration and number of data points. The performance of the best performing overall model, the "Enhanced Balanced Random Forest Anomaly Classifier", as a function of number of data points in the light curve is shown in Fig. 12. From this graph, we see that the misclassification rate of both stable and unstable light curves drops once the number of datapoints N falls below 75. Assuming a median lag time of 10 seconds in between data points, this result would correspond to a light curve observation on the order of approximately 8 to 12.5 minutes in duration.

Our qualitative analysis of the light curves for which this classifer had higher overall MSE showed that these light curves shared three major characteristics. These included: (1) unstable light curves for which less than a single tumbling rotation occurs within the period of observation, (2) aliased tumbling light curves, and (3) stable light curves for which a glint occurred within a short phase angle range. Issues (1) and (3) seemed to have a limited effect on the classifier performance for light curves of duration of $\geq$ 15 minutes or approximately $N = 75$ data points in length. Our future goal is to work with providers to translate these sorts of takeaways into provider collection plans that maximise collection quality for the spin stability assessment problem.

7. FUTURE DIRECTION: SPIN RATE RETRIEVAL

After observing the autoencoder's ability to retrieve time-dependent rotational trends in the light curves, we wanted to test the ability of our autoencoder to aid in spin-rate retrieval of tumbling satellites. The potential of RNNs for

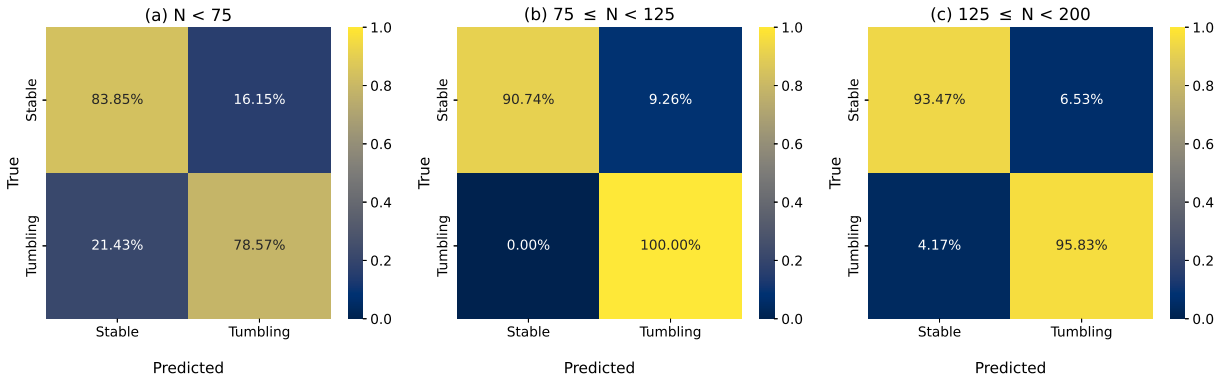


Fig. 12: The "Enhanced Balanced Random Forest Anomaly Classifier" model's classification performance as a function of the length of the light curve in number of data points, N.

this task has shown strong promise. For example, Naul et al demonstrated that feeding their encoder's embedding layers to dense layers resulted in performance that was on par with the performance of Lomb-Scargle periodograms in retrieving spin-rate [27].

In order to test spin-rate retrieval capabilities, we generated unevenly sampled light curves using our previously created light curve simulation tools [3] and fed these synthetic light curves to our autoencoder that was trained on real light curve data. An example of the results of feeding one such light curve to our autoencoder is shown in Fig. 13. The true spin-rate of the satellite for this simulation was $\omega = 0.15$ Hz, and the simulated visual magnitude uncertainty was 0.4. The responses of the embedding layers to this light curve were very interesting. For example, consider embedding #10 which responds to the periodicity of the light curve until a phase angle of approximately $\phi = 17.5^{\circ}$ and then flatlines. In future studies, we will try to understand if this is a learned geometry-dependent response.

It can also be seen from Fig. 13 that (1) the embedding signals have a lower amount of noise in terms of their relative magnitudes, and (2) low-frequency signals in the light curve data are reduced. In short, we believed that the encoder was behaving as a type of data cleaning operator on the light curve. When feeding the raw light curve data to a Lomb-Scargle periodogram, we found that the algorithm returned the highest power for a low frequency of approximately 0.06 Hz, suggesting that long-term trend information must be removed via pre-processing in order to obtain accurate results.

In order to test the data cleaning ability of the encoder, we fed each embedding signal to the same Lomb-Scargle periodogram. The results are shown in Fig. 14. We can see that many of the layers do an excellent job of reducing the influence of low-frequency trends on the spin-rate estimation. Many of the layers correctly predict the true spin-rate without any additional pre-processing required. We are currently working on expanding this capability to incorporate information such as longitudinal phase angles into the spin-rate estimation process.

8. CONCLUSIONS

The ever-growing volume of light curve data being produced and pushed onto repositories such as the Unified Data Library (UDL) calls for the development of automated machine learning frameworks to aid human analysts in performing satellite assessment. A major challenge that machine learning framworks must deal with is that these light curves are often unevenly sampled in regards to the time lag in between data points and the overall number of data points in the time series.

In this study, we made efforts towards addressing these gaps on geostationary light curves that were pulled from the UDL. Specificially, we developed an autoencoder-based anomaly detection algorithm that provides spin-stability assessments on light curves as short as 5 minutes in duration with up to a 91% score in balanced accuracy. The autoencoder was trained to reconstruct noisy light curves with explicit knowledge of the phase angle and visual magnitude uncertainty of each data point in order to overcome common issues such as data dropouts and weather fluctuations.

Our analysis of this architecture's predictions showed that the reconstruction prediction accuracy served as an "anomaly

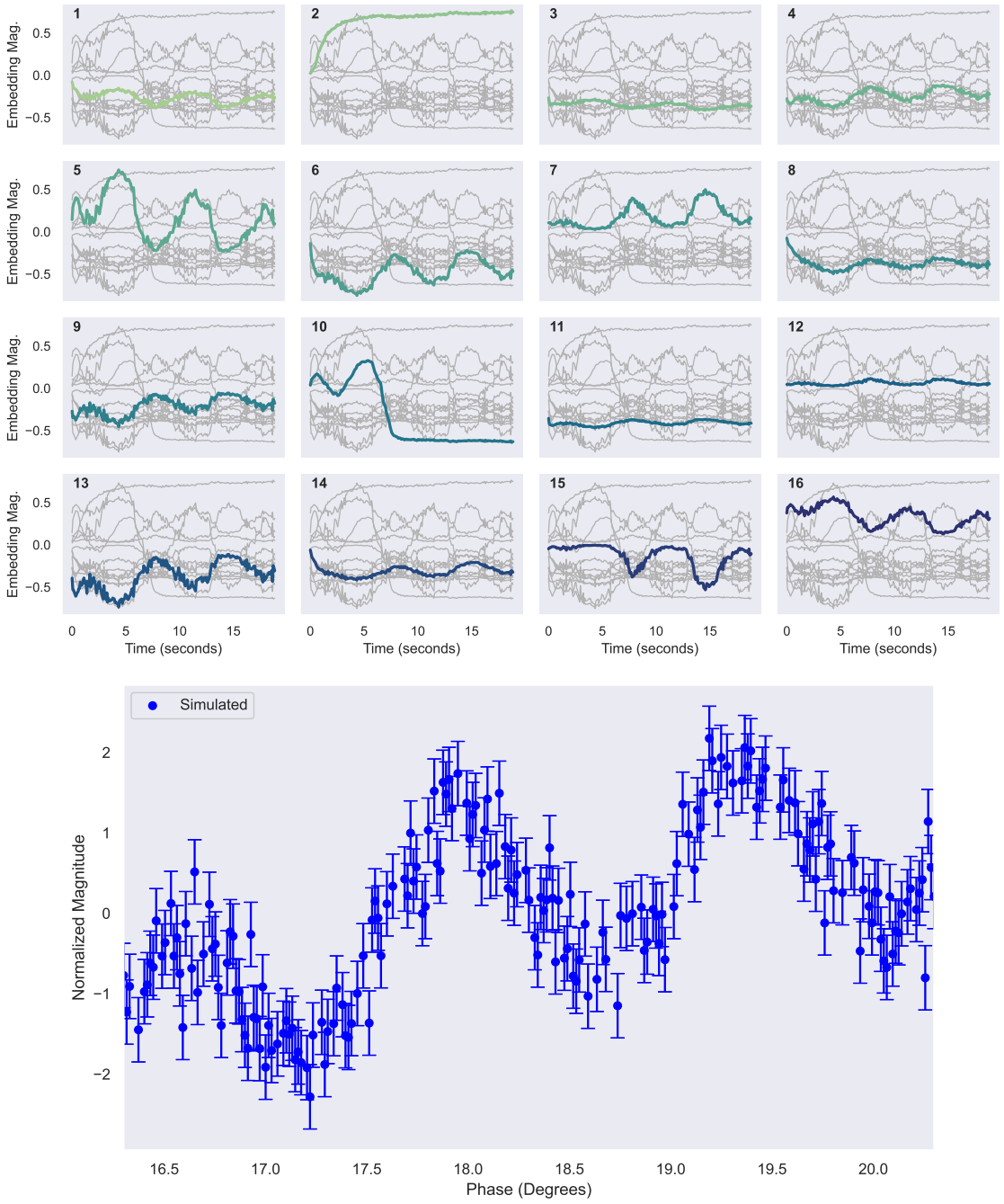


Fig. 13: Example of embeddding responses to a noisy simulated light curve for which the spin rate is known (Top) Responses of unique embedding layers to the light curve. (Bottom) The raw data input to the autoencoder.

score" for discriminating unstable light curves from stable light curves. In particular, we found that the autoencoder did not reconstruct aliased signals properly, suggesting that a feature representation could be developed to flag aliased light curves using the embedding feature vectors of the autoencoder.

Based on these insights, the autoencoder's uncertainty-weighted mean reconstruction error was combined with several expert-derived features (Table 4) as a feature embedding for a Balanced Random Forest (RF) classifier. The Balanced RFs that were trained using the autoencoder's reconstruction error metric outperformed those trained solely using

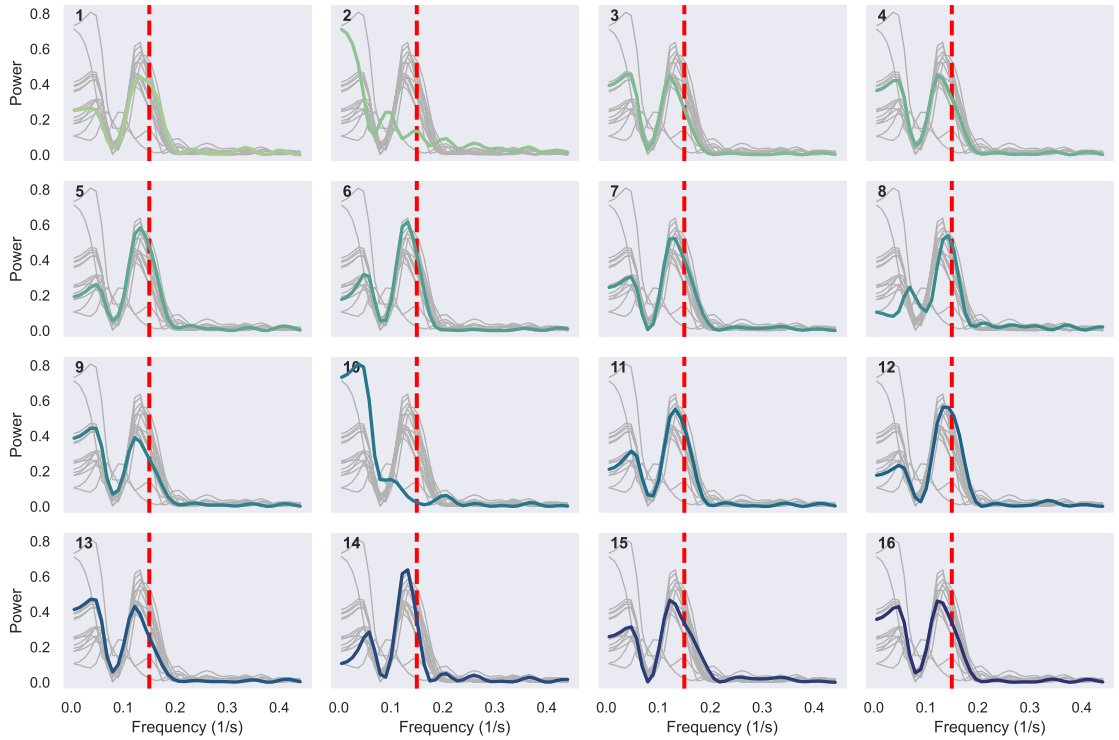


Fig. 14: Lomb-Scargle power curves for the various embedding vectors of the autoencoder with the true frequency of 0.15 Hz shown in red.

expert level features by 9±3.5% in terms of Matthews Correlation Coefficient (MCC) and 8.5±3.7% in terms of F1-score, with a significance level of $p < 1 \times 10^{-5}$.

This promising initial effort shows that machine learning frameworks that are trained with explicit knowledge of the collection cadence of the source telescope can make highly accurate spin-stability predictions on irregularly sampled light curves. However, there are still many directions that we can take this research in.

Other researcher's results suggest that the spin-stability classification accuracy and reconstruction accuracy can be improved by utilizing "period-folding" in order to provide the classifier with an estimate of the space object's spin rate [27]. Therefore, in our next study we will extend this autoencoder framework towards spin rate retrieval. Our future work will also analyse if other features besides the weighted light curve reconstruction error can be utilized to improve the classification accuracy. This will include searching for any periodic behaviors in the embedding vectors or the output reconstruction.

9. ACKNOWLEDGEMENTS

We would like to thank Air Force Research Laboratory (AFRL) for sustaining the UDL program that has allowed us to test our models for spin-stability classification on real satellite observations. We would also like to thank ExoAnalytic Solutions for sharing the UDL data that allowed us to perform the analyses in this study.

We would additionally like to thank Joe Gerber (Joint Task Force Space Defense and KBR Aerospace) and Riley Burfield (Tech 7) for assistance with incorporating the model into the December 2021 SACT event and with labeling the Machine Learning datasets used in our evaluation routines.

Finally, we would like to thank GTRI's Internal Research and Development (IRAD) funds for supporting this work.

10. REFERENCES

[1] Antonio L Alfeo, Mario GCA Cimino, Giuseppe Manco, Ettore Ritacco, and Gigliola Vaglini. Using an autoencoder in the design of an anomaly detector for smart manufacturing. *Pattern Recognition Letters*, 136:272–278, 2020.

[2] Gregory Badura, Christopher R Valenta, Brian C Gunter, and Brian Shoffeitt. Multi-scale convolutional neural networks for inference of space object attitude status from detrended geostationary light curves. In *31st AAS/AIAA Space Flight Mechanics Meeting*, 2021.

[3] Gregory P Badura, Christopher R Valenta, and Brian Gunter. Convolutional neural networks for inference of space object attitude status. *The Journal of the Astronautical Sciences*, pages 1–34, 2022.

[4] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[5] Davide Chicco and Giuseppe Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21(1):6, 2020.

[6] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[7] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[8] Rita Cognion. Observations and modeling of geo satellites at large phase angles. *AMOS Proceedings*, 2013.

[9] Phillip M Cunio, Michael Bantel, Brien R Flewelling, William Therien, Mark W Jeffries Jr, Monica Montoya, Rhett Butler, and Douglas Hendrix. Photometric and other analyses of energetic events related to 2017 geo rso anomalies. In *Proceedings of the Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference*, 2017.

[10] Phan Dao, Kristen Haynes, Stephen Gregory, Jeffrey Hollon, Tamara Payne, and Kimberly Kinateder. Machine classification and sub-classification pipeline for geo light curves. In *Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference*, 2019.

[11] Arka Daw, Anuj Karpatne, William Watkins, Jordan Read, and Vipin Kumar. Physics-guided neural networks (pgnn): An application in lake temperature modeling. *arXiv preprint arXiv:1710.11431*, 2017.

[12] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

[13] Sandra Erwin. https://spacenews.com/bluestaq-wins-280-million-space-force-contract-to-expand-space-data-catalog, Mar 2021.

[14] Carolin Frueh, Hauke Fielder, and Johannes Herzog. Heuristic and optimized sensor tasking observation strategies with exemplification for geosynchronous objects. *Journal of Guidance, Control, and Dynamics*, 41(5):1036–1048, 2018.

[15] Roberto Furfaro, Richard Linares, and Vishnu Reddy. Shape identification of space objects via light curve inversion using deep learning models. In *AMOS Technologies Conference, Maui Economic Development Board, Kihei, Maui, HI*, 2019.

[16] Joseph D. Gerber, Yann Picard, Melrose Brown, Jason Held, Frederic Pelletier, and Timothy Fuller. Sprint advanced concept training (sact): A renaissance in collaborative international space operations. In *Proceedings of the Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference*, 2020.

[17] Kristen Haynes, Jeffrey Hollon, Kimberly Kinateder, Victoria Carone, and Tamara Payne. Automated multi-sensor data fusion using the unified data library. *Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference*, 2021.

[18] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[19] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4):917–963, 2019.

[20] Emma Kerr, Gabriele Falco, Nina Maric, David Petit, Patrick Talon, Elisabeth Petersen, Chris Dorn, Stuart Eves, Noelia Sánchez-Ortiz, Raul Dominguez Gonzalez, et al. Light curves for geo object characterisation. In *8th European Conference on Space Debris*, 2021.

[21] Paul W Kervin, Doyle Hall, Mark Bolden, and Joe Toth. Phase angle: What is it good for. In *Proceedings of the Advanced Maui Optical Space and Surveillance Technologies Conference, Maui, HI, USA*, pages 14–17, 2010.

[22] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017.

[23] Richard Linares and Roberto Furfaro. Space object classification using deep convolutional neural networks. In *2016 19th International Conference on Information Fusion (FUSION)*, pages 1140–1146. IEEE, 2016.
[24] Benjamin Lindemann, Fabian Fesenmayr, Nasser Jazdi, and Michael Weyrich. Anomaly detection in discrete manufacturing using self-learning approaches. *Procedia CIRP*, 79:313–318, 2019.
[25] Keiran McNally, Diego Ramirez, Alfredo M Anton, Duncan Smith, and James Dick. Artificial intelligence for space resident objects characterisation with lightcurves. In *8th European Conference on Space Debris*, 2021.
[26] Ian McQuaid, Laurence D Merkle, Brett Borghetti, Richard Cobb, and Justin Fletcher. Space object identification using deep neural networks. In *The Advanced Maui Optical and Space Surveillance Technologies Conference*, page 5, 2018.
[27] Brett Naul, Joshua S Bloom, Fernando Pérez, and Stéfan van der Walt. A recurrent neural network for classification of unevenly sampled variable stars. *Nature Astronomy*, 2(2):151–155, 2018.
[28] Matthew Phelps, J Zachary Gazak, Thomas Swindle, Justin Fletcher, and Ian Mcquaid. Inferring space object orientation with spectroscopy and convolutional networks. *AMOS (Sept. 2021)*, 2021.
[29] Juan D Rodriguez, Aritz Perez, and Jose A Lozano. Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE transactions on pattern analysis and machine intelligence*, 32(3):569–575, 2009.
[30] Vijay K Rohatgi and AK Md Ehsanes Saleh. *An introduction to probability and statistics*. John Wiley & Sons, 2015.
[31] Ewan Schafer. *Stereoscopic light curve analysis of space debris objects*. PhD thesis, Julius Maximilians Universität Würzburg, Luleå Tekniska Universitet, 2017.
[32] Chang Wei Tan, François Petitjean, Eamonn Keogh, and Geoffrey I Webb. Time series classification for varying length series. *arXiv preprint arXiv:1910.04341*, 2019.
[33] Xia Wang, YuRong Huo, YuQiang Fang, Feng Zhang, and Yifan Wu. Arsrnet: accurate space object recognition using optical cross section curves. *Applied Optics*, 60(28):8956–8968, 2021.
[34] Jinlong Wu, Xiaolong Yin, and Heng Xiao. Seeing permeability from images: fast prediction with convolutional neural networks. *Science bulletin*, 63(18):1215–1222, 2018.
[35] Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*, 2017.
[36] Yuexin Zhang. A fusion methodology to bridge gps outages for ins/gps integrated navigation system. *IEEE access*, 7:61296–61306, 2019.

11. APPENDIX

A. EXPERT FEATURES USED IN STUDY

In this section, we show a table of the expert-derived features that were used in the baseline model of this study. The definitions of each metric come from the original study by Dao et al on classifying geostationary satellite stability using a random forest model [10]. These features are shown in Table 4. Note that we only used features which were deemed appropriate for short-duration light curves as opposed for full-night light curves.

B. MACHINE LEARNING EVALUATION FEATURES

B.1 F1 Score

The precision of a binary classifier defined by considering the number of true-positives, TP, and false positives, FP, predicted by the classifier and is denoted by : $precision = TP/(TP+FP)$. The recall of a classifier, on the other hand, takes into account the number of false-negatives, FN, and can be written as: $recall = TP/(TP+FN)$. The F1-score can be considered as a harmonic mean of precision and recall [5]:

$$F1 = 2\left(\frac{precision \times recall}{precision + recall}\right) \tag{9}$$

The $F1$ metric ranges from 0 to 1, where the minimum is reached when all the positive samples are mis-classified, and the maximum is reached for perfect classification. F1 score is independent from the number of samples correctly classified as negative, which some researchers have stated as a notable flaw of the metric [5].

Table 4: Expert level features used in training the Balanced Random Forest classifier [10].

Metric	Description
Average Visual Magnitude	The average visual magnitude within a light curve time series.
Standard Deviation of Visual Magnitudes	The standard deviation of the observed visual magnitude values.
Range of Visual Magnitudes	The absolute value of the difference between maximum and minimum of the observed visual magnitude values.
Number of Peaks Per Phase Angle	The number of peaks in a light curve divided by total range of phase angle values.
${R_{linear}}^2$	Coefficient of determination for a linear regression of phase angle and visual magnitude
${R_{quadratic}}^2$	Coefficient of determination for a quadratic regression of phase angle and visual magnitude
${R_{cubic}}^2$	Coefficient of determination for a cubic regression of phase angle and visual magnitude
Skewness	The skewness as defined by the Fisher-Pearson coefficient. For normally distributed data, this metric is approximately zero.
Kurtosis	The kurtosis as defined by Fisher's method where 3.0 is subtracted from result. For normally distributed data, this evaluates to approximately zero.

B.2 Matthews Correlation Coefficent Score

The Matthews Correlation Coefficient (MCC) has been shown to be robust to unbalanced datasets [5]. In a binary classification scenario, it generates a high score only if the predictor is able to correctly predict the majority of positive data instances and the majority of negative data instances. It ranges in the interval [1,+1], with extreme values –1 and +1 achieved in case of perfect mis-classification and perfect classification, respectively, while MCC=0 signifies random chance [5]. The equation for MCC can be written as:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{10}$$

Where the incorporation of the True Negative, TN, shows that MCC will only generate a high score if the classifier does well in all four quadrants of the confusion matrix.

B.3 Balanced Accuracy

Balanced accuracy is computed using two standard machine learning metrics of *recall* or true positive rate and $specificity$ or true negative rate. The equation for recall was previously defined above. The equation for $specificity$ can be written as $specificity = TN/(TN+FP)$. Balanced accuracy is then the arithmetic mean of these two metrics:

$$BA = \frac{recall + specificity}{2} \tag{11}$$