# Deep Learning Techniques for Eclipsing Binary Light Curves Classification

**Jinbo Fu**[1]
**Xinhao He**[1]
**Tuan Yi**[1]

[1]Department of Astronomy, Xiamen University, Xiamen, Fujian 361005, Peoples Republic of China
fujinbo@stu.xmu.edu.cn

## Abstract

In a binary system, two stars rotate together around a common center of mass. Long-term photometric observations have revealed to astronomers very rich and diverse types of periodic variability associated with different parameters of binary systems. The classification of this zoo of variable stars across the sky is a challenging, useful, and fascinating task that may lead to new discoveries such as compact objects in binary systems. In this work, we use the photometric time series of binary systems as input to classify the systems into several typical classes without any pre-assumption of period. We show that the classifier trained on the ASAS-SN survey data effectively characterizes and classifies the stellar parameters.

## Introduction

Eclipsing binary systems are binary systems with orbital planes so close to the observer's line of sight (the inclination $i$ of the orbital plane to the plane orthogonal to the line of sight is close to $90\,deg$) that the components periodically eclipse each other. Consequently, the observer finds changes in the system's apparent combined brightness with the period coincidence with that of the components' orbital motion.

In 1783, John Goodricke showed that Algol's variations were periodic: it gets about 2 magnitudes fainter than normal every 68.8 hours. He speculated that either Algol had an unseen body in orbit about it with a period of 68.8 hours, or that Algol had dark spots which came into view every 68.8 hours (Goodricke 1783). H. C. Vogel showed in 1890 that Algol was a spectroscopic binary with a period of 68.8 hours (Vogel 1890). Furthermore, he showed that the primary star was receding just before the eclipse, and approaching just after the eclipse. By about 1840, a few dozen variable stars were known, these were discovered visually. Presently, the ASAS-SN Variable Stars Database (AVSD) contains 216,446 confirmed variable stars, 666,502 records of light curves, and most were discovered in large-scale CCD photographic patrols (Shappee et al. 2014; Jayasinghe et al. 2019). The vast majority of these have not been studied in any detail.



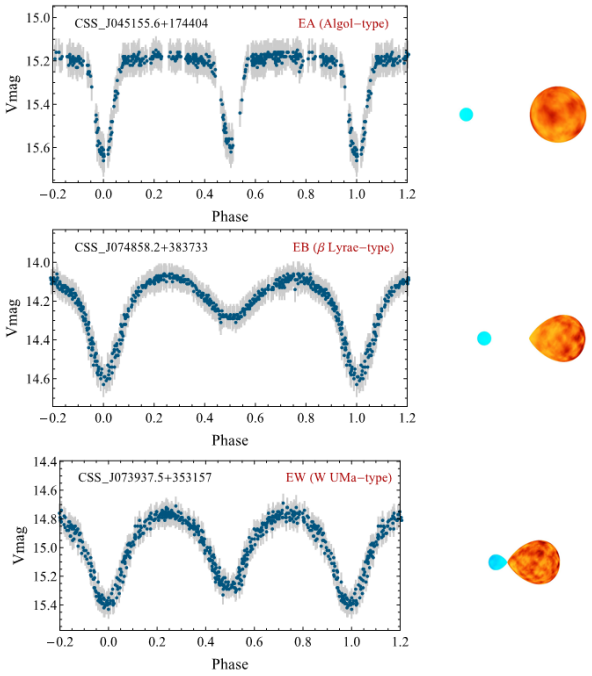


Figure 1: Left panels: Example of different eclipsing binary light curve types. Right panels: the binary configuration that corresponds to each type on the left, respectively.

Historically, there are three basic classes of eclipsing light curves, based solely on the overall light curve shape (Figure 1):

- **EA** Type: An eclipsing variable of the Algol type. There are well-defined eclipse(s) and little variation in between. EAs are detached binary systems with two stars well separated; each one is within its Roche Lobe;
- **EB** Type: An eclipsing variable of the $\beta$ Lyrae type. There are well-defined eclipse(s) but considerable variation in between. EBs are semi-detached binary systems with one of the star fills its Roche Lobe while the other one does not;
- **EW** Type: An eclipsing variable of the W Ursae Majoris type. There are no well-defined eclipse(s) and little distinction between the eclipse and out-of-eclipse phases.

EWs are contact binary systems with two stars both fills their Roche-lobes.

Many binary systems will also show additional features that complicate the interpretation of the light curve, mostly due to starspots (bright and/or dark) or to mass transfer streams. There are up to 40 different types in AVSD.

In reality, because of weather, observational plans, and many other uncontrollable conditions, the cadence of long-term observations are almost always irregular, with gaps in between observation windows and non-uniform sampling (Figure 2). This fact posts a big challenge to the classification task. Traditionally statistical features have been derived from the light curves to do follow-up classification. The features include standard statistical measures like median, skew, kurtosis and specialized domain knowledge-based ones such as "fading profile of a single-peaked fast transient". The standard features do not carry special powers for classifying a varied set of objects. The designer features are better for specific classes, but carry a bias that does not necessarily translate to a broader set classification. The accuracy of the task is greatly improved by machine learning, but mostly requires computing the period of the binary star system as input to the classifier model. The algorithms used to calculate the period, such as the Lomb-Scargle periodogram are usually based on a priori statistics and therefore introduce uncertainty into the results (Zechmeister and Kürster 2009). Convolutional neural network (CNN) trained with periodically folded data also lose the time-domain information of the original data.

Here we propose a classifier based on the Gated Recurrent Unit (GRU) recurrent neural network (RNN) to classify the light curve without assuming about the orbital period of the binary star. RNN is suitable of identifying features in time series data and is possible to learn the characterization of periodic information. We discuss related work in Section 2, then describe data collection and preprocessing in Section 3. The theoretical description and detailed architecture of the classifier are located in Section 4 and Section 5, respectively. The performance of the classifier is discussed in Section 6, then concluded in Section 7.

## Related works

Machine learning has been widely used in the past for the task of light curve identification and clustering. A refined classifier for eclipsing binaries based on pattern regression and Bayesian network found an overall accuracy improvement of 12.1% compared above a simple 50-30-7 multilayer perceptron (Sarro, Sánchez-Fernández, and Giménez 2006). Debosscher et al. trained both single-stage SVM classifiers and sequential classifiers. The unique automated supervised classifier for each of the 29 classes with sufficient stars uses the Bayesian Model Averaging (BMA) instead of the maximum likelihood estimate provided by the error backpropagation algorithm and produces an average correct classification rate of 70%. The SVM optimized by grid search and 10-fold cross-validation got 50% correct identifications (Debosscher et al. 2007). Modak, Chattopadhyay, and Chattopadhyay applied k-medoids clustering with complexity invariance distance (CID) to the light curves and gave rise to two physically interpretable groups of Eclipsing binaries, which has outperformed the established methods in average silhouette width (ASW) (Modak, Chattopadhyay, and Chattopadhyay 2020).

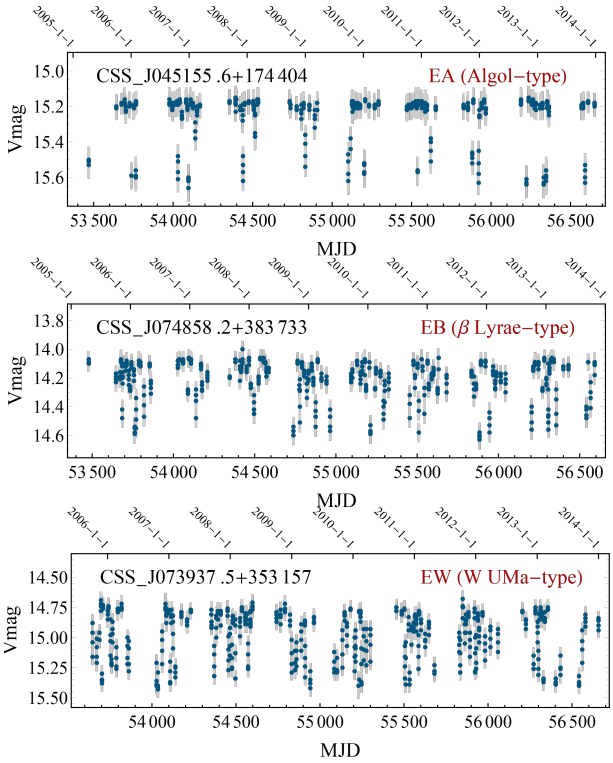


Figure 2: The original light curve from Catalina Sky Server, correspond with the sources in Figure 1.

The best-performing and widely applied algorithm is the Random Forest Algorithm (RFA), which has been used in The Catalina Sky Survey (CSS), Kepler mission $K2$, Hipparcos periodic variable stars and AVSD. It achieves an $F_1$ score of 96.25% in the binary classification and 96.83% in the eight-class classification (Neira et al. 2020). Artificial neural network (ANN, Karpenka, Feroz, and Hobson 2013), CNN (Cabrera-Vives et al. 2017), recurrent convolutional neural network (RCNN, Carrasco-Davis et al. 2019), and long short-term memory (LSTM, Charnock and Moss 2017) RNN both are common approaches to understand what physical properties are most related to the light curve (Hinners, Tat, and Thorp 2018).

## Data collection and preparation

This work investigates using real light curve observations taken from The All Sky Automated Survey for SuperNovae (ASAS-SN). For this task, the training data was developed using segments of 63,645 measurement samples taken from the ASAS-SN dataset for objects with light curve type information. Their ASAS-SN numbers are unique associated with the objects in the ASAS-SN dataset, and from its names, class information can be extracted. Four classes, EA,

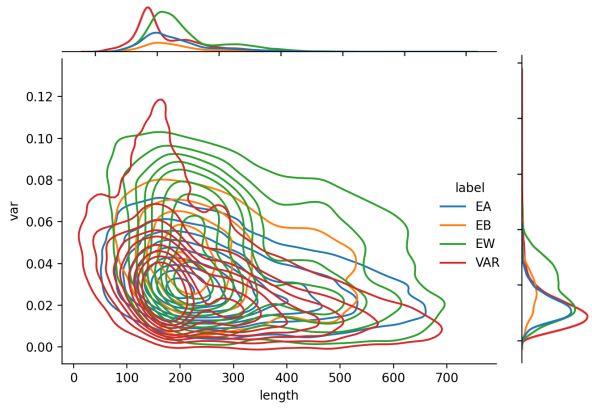


Figure 3: The distribution of sequence length and variance after rescale for each type of light curves. It shows the length and sample size of different types of sequences vary greatly.

EB, EW, and VAR (not fall into any of the above categories), are used in this work.

To create a good and clean training sample, for each sample of light curve $\{t_i, m_i, \delta m_i\}$, where $t_i$ is the time point of the data record, $m_i$ is the corresponding observed magnitude at this time point and $\delta m_i$ is the observation error, we made a number of cuts before processing the light curves. The selection criteria is described as follows.

*Classify probability.* To ensure the accuracy of the dataset labeling, we selected only targets with a classification confidence level of 1. For VAR type data, this indicator is reduced to $0.5$ in order to ensure that enough VAR data samples enter the training set, and VAR data is not easily confused with other types. The sample sizes is 12,618 for EA type, 5,995 for EB type, 25,193 for EW type, and 19,839 for VAR type. It should be emphasized that the unevenness of the sample size is an important factor affecting classification effect. The distribution of selected data samples are shown in Figure 3.

*Remove outliers.* The light curves are measured in flux units, as is expected for the ASAS-SN difference imaging pipeline. The primitive data have a significant fraction of the observations being $5 - 10$ $\sigma$ outliers. These outliers are intended to replicate the difference image analysis artifacts, telescope CCD deficiencies, and cosmic rays seen in observational data. We perform "sigma clipping" to reject these outliers. We do this by rejecting photometric points with flux uncertainties that are more than $3\sigma$ from the mean uncertainty in each record, and iteratively repeat this clipping $5$ times.

*Panning light curves.* One of the key differences in this work compared to previous light curve classification approaches is our ability to provide time-varying classifications. The Key to computing this, is just make the observation time point as another dimension of the input of our model. Cause the observed durations of light curves for some sources may be very long and are recorded using the Modified Julian Date (MJD) formation, we move the time series data as a whole to $t_0 = 0$ to avoid the effect of time data on model convergence.

*Rescale light curves.* Different types of binary stars deride to different level of magnitude. To ensure that the difference in luminance does not affect the classification accuracy, all data of magnitude observations were applied to the follow function

$$m_i = \frac{m_i - m_{min}}{m_{max} - m_{min}}$$

to scale into $m_i \in [0, 1]$, where $\bar{m}$ represents the average of the photometric data.

## Model

In this work, we train a deep neural network (DNN) to map the light curve data of an individual object $s$ onto probabilities over classes $c = \{\text{EA, EB, EW, VAR}\}$. The DNN models a function that maps an input time-varying light curve matrix, $I^{st}$, for object $s$ up to a discrete time $t$, onto an output probability vector,

$$y^{st} = f_t(I^{st}; \theta),$$

where $\theta$ are the parameters of our DNN architecture. We define the input $I^{st}$ as an $n \times 3$ matrix representing the light curve up to a time point $t_i$, the magnitude $m_i$ and the observation error $\delta m_i$. The output $y^{st}$ is a probability vector with shape $4 \times 1$, where each element $y_c^{st}$ is the model's predicted probability of each class c, such that $y_c^{st} \geq 0$ and $\Sigma_{c=1}^{4} y_c^{st} = 1$.

First, to quantify the discrepancy between the model probabilities and the class labels we define a weighted categorical cross-entropy,

$$H_w(Y^{st}, y^{st}) = -\Sigma_{c=1}^{m+1} w_c Y_c^{st} \log(y_c^{st}),$$

where $w_c$ is the weight of each class, $Y^{st}$ is a one-hot encoded vector label of shape $4 \times 1$ for the true light curve class.

To counteract imbalances in the distribution of classes in the data set which may cause more abundant classes to dominate in the optimization, we define the weight for each class $w_c$ as

$$w_c = \frac{p_c}{\Sigma_{c=1}^{4} p_c},$$

where $p_c = 1 - \frac{N_c}{N}$ and $N_c$ is the number of times a particular class appears in $N$ training set.

We define the global objective function as

$$obj(\theta) = \Sigma_{s=1}^{N} H_w(Y^{st}, y^{st}),$$

where we sum the weighted categorical cross-entropy over all $N$ light curves in the training set. To train the DNN and determine optimal values of its parameters $\hat{\theta}$, we minimize this objective function with the sophisticated and commonly used Adam gradient descent optimizer. The model $f_t(I^{st}; \hat{\theta})$ is represented by the complex DNN architecture illustrated in Figure 4 and is described in the following section.

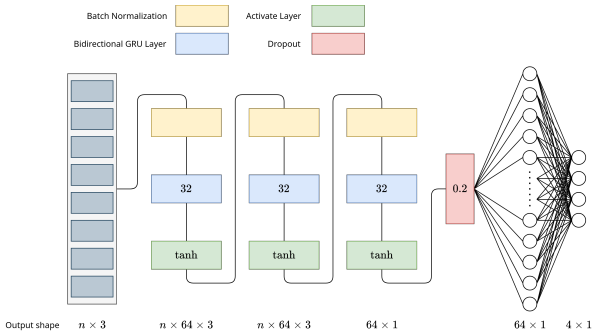


Figure 4: The architecture of our network.

| Parameter | Value Range | Step |
|---|---|---|
| Learning rate | 0.001 | Dynamic adjustment |
| GRU units | 128, 64, **32**, 16 | - |
| Batch size | **512**, 256, 128 | - |
| Sequences padding maxlen | 100 - 1500 | 100 |
| Dense units | 128, **64**, 32, 16 | - |
| Activation | tanh, ReLU, softmax | - |
| Initializer | **GlorotUniform**, **Orthogonal**, LeCunUniform, LeCunNormal HeUniform, HeNormal | - |
| Dropout rate | 0 - 0.4 | 0.05 |
| Batch normalization | **True** / False | - |
| Bidirectional | **True** / False | - |

Table 1: Hyper-parameters selection for our model. The chosen values are marked in bold.

## Network Architecture

RNNs, such as LSTM and GRU networks have been shown to achieve state-of-the-art performance in many benchmark timeseries and sequential data applications (Bahdanau, Cho, and Bengio 2014; Sutskever, Vinyals, and Le 2014; Che et al. 2018). Its success in these applications is due to its ability to retain an internal memory of previous data, and hence capture long-term temporal dependencies of variable-length observations in sequential data. We extend this architecture with bi-directional to pass information both forwards and backwards through the neural network representation of the light curve, and hence preserve information on both the past and future at any timestep. The deep neural network (DNN) is illustrated in Figure 4. We have developed the network with the high level Python API, Keras (Chollet et al. 2018), built on the recent highly efficient TensorFlow machine-learning system (Abadi et al. 2016). We describe the architecture in detail here.

*Input.* As detailed in before, the input is an $n \times 3$ matrix. However, as we are implementing a sequence classifier, we can consider the input at each time point as being a vector of length $3 \times 1$.

*GRU Layer.* Gated recurrent units are an improved version of a standard RNN and are a variation of the LSTM (Chung et al. 2014). We have selected GRUs instead of LSTMs in this work, as they provide appreciably shorter overall training time, without any significant difference in classification performance. Both are able to capture long-term dependencies in time-varying data with parameters that control the information that should be remembered at each step along the light curve. We use the first GRU layer to read the input sequence one timestep at a time and encode it into a higher-dimensional representation. The second GRU layer has a $64 \times 3$ matrix as output similarly and the third GRU layer output a $64 \times 1$ vector. We use bi-directional GRUs that enable both information from previous and future timesteps are encoded.

*Batch Normalization.* We then apply Batch Normalization (Ioffe and Szegedy 2015) to each GRU layer. This acts to improve and speed up the optimization while adding stability to the neural network and reducing overfitting. While training the DNN, the distribution of each layer's inputs changes as the parameters of the previous layers change. To allow the parameter changes during training to be more stable, batch normalization scales the input.

*Dropout.* We also implement dropout regularization between each layer of the neural network to reduce overfitting during training. This is an important step that effectively ignores randomly selected neurons during training such that their contribution to the network is temporarily removed. This process causes other neurons to more robustly handle the representation required to make predictions for the missing neurons, making the network less sensitive to the specific weights of any individual neuron. We set the dropout rate to 20% of the neurons present in the previous layer.

*Dense Layer.* A dense layer is the simplest type of neural network layer. It connects all 64 neurons in the previous layer, to the $4 \times 1$ neurons in the output layer. As this is a classification task, the output is a vector consisting of all $4$ light curve classes.

*Activation function.* As with any neural network, each neuron applies an activation function $f(\cdot)$ to bring nonlinearity to the network and hence help it to adapt to a variety of data. For feed-forward networks it is common to make use of Rectified Linear Units (ReLU, Nair and Hinton 2010) to activate neurons.

We reiterate that the overall architecture is simply a function that maps an input $n \times 3$ light curve matrix onto an $4 \times 1$ softmax probability matrix indicating the probability of each light curve class. In order to optimize the parameters of this mapping function, we specify a weighted categorical cross-entropy loss-function that indicates how accurately a model with given parameters matches the true class for each input light curve. We minimize the objective function using the commonly used, but sophisticated stochastic gradient descent optimizer called the Adam optimizer (Kingma and Ba 2014). As the class distribution is inevitably uneven, we prevent bias towards overrepresented classes by applying class-dependent weights while training.

The several layers in the DNN create a model that has 266,000 free parameters. As we feed in our training set in batches of 512 light curves at a time, the neural network updates and optimizes these parameters. Ideally, we would like to ensure that the model we build both accurately captures regularities in the training data while simultaneously generalizing well to unseen data. While we implement regularization layers (dropout) to try to prevent overfitting, we also monitor the performance of the classifier on the training and testing sets during training. In particular, we ensure

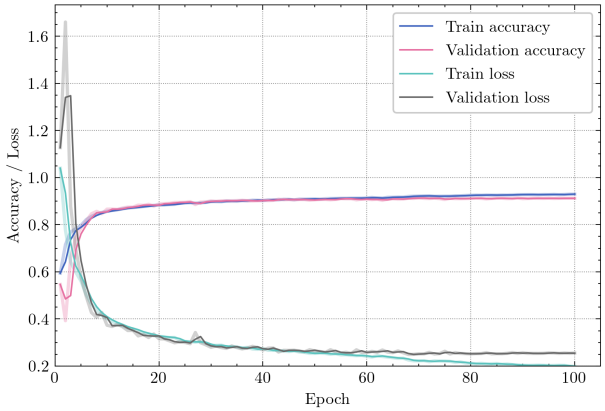


Figure 5: The accuracy and loss in first 100 training epoch.

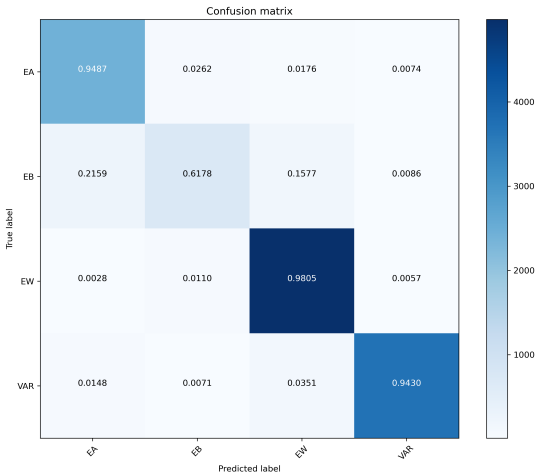


Figure 6: The confusion matrix for DNN.

that we do not run the classifier over so many iterations that the difference between the values of the objective function evaluated on the training set and the testing set become significant.

We performed a broad grid-search of 10 of our DNN hyper-parameters which are shown in Table 1. After testing all different setup parameters, we found that there was about 10% variation on the overall accuracy.

## Result

We trained our DNN on 80% of this set and tested its performance on the remaining 20%. The data was preprocessed using the methods outlined before. To assess the performance of our model, we make use of several metrics. The most obvious metric is simply the accuracy, that is, the ratio of correctly classified light curve in each class to the total number of light curve in each class. At each epoch along every light curve in the testing set, we select the highest probability class and compare this to the true class. The weighted cross-entropy of the predicted class and the true class, which is just the loss in this task, is also used as another metric. Here we obtained the prediction accuracy and loss as a function of epoch. This is plotted in Figure 5.

In Figure 6, we plot the normalized confusion matrices for different types of light curve. The overall classification performance is 91.37%, as expected, slightly worst at the EB type of light curves due to the lack of samples. However, the performance for the other types is particularly promising for our ability to identify different types at early times to gather a well-motivated follow-up candidate list.

The overall accuracy of our DNN is lower than the SOTA model of 96.25%, cause the input light curve of the SOTA model has been preprocessed by period folding algorithm such as the Lomb-Scargle periodogram. Although we make time-point series as one column of the input matrix, our DNN does not seem to have fully learned such an algorithm to auto-fold observation time-points. The fact that the EB type samples are few that DNN does not learn enough classification information is also the reason for the poor overall performance.

## Conclusion

We have detailed the development of a new binary stellar light curve classifier, which is well suited for the millions of light curves per night that ongoing and upcoming wide-field surveys such as ZTF and LSST will produce. The key advantages that distinguish our approach from others in the literature are:

- Our architecture combined with a diverse training set allows us to identify 4 different light curve classes, despite low S/N and limited observation times.
- We do not require user-defined feature extraction and time folding before classification, and instead use the processed light curves as direct inputs.
- Our algorithm is designed from the outset with speed as a consideration, and it can classify the tens of thousands of events that will be discovered in each LSST image within a few seconds.

While we designed this model primarily for eclipsing binary light curves classification, the flexibility of our architecture means that it is also useful for photometric classification with any available types of the light curves. We have presented detailed confusion matrices, accuracy and loss curves for all the classes represented in our training set, evaluated accuracy across the 4 classes is up to 91.37%.

In future work, we plan on applying this method on LSST simulations to help to inform how changes in observing strategy affect photometric classifications and find some compact objects such as black holes and dwarf stars. We will analyze further the physical meaning behind the statistical results and investigate possible causes and implications. Overall, our DNN provides a novel and effective method of classifying eclipsing binary light curves and providing prioritized follow-up candidates for the new era of large scale stellar surveys.

## References


Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, 265–283.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .

Cabrera-Vives, G.; Reyes, I.; Förster, F.; Estévez, P. A.; and Maureira, J.-C. 2017. Deep-HiTS: Rotation Invariant Convolutional Neural Network for Transient Detection. 836(1): 97. doi:10.3847/1538-4357/836/1/97.

Carrasco-Davis, R.; Cabrera-Vives, G.; Förster, F.; Estévez, P. A.; Huijse, P.; Protopapas, P.; Reyes, I.; Martínez-Palomera, J.; and Donoso, C. 2019. Deep learning for image sequence classification of astronomical events. *Publications of the Astronomical Society of the Pacific* 131(1004): 108006.

Charnock, T.; and Moss, A. 2017. Deep Recurrent Neural Networks for Supernovae Classification. 837(2): L28. doi:10.3847/2041-8213/aa603d.

Che, Z.; Purushotham, S.; Cho, K.; Sontag, D.; and Liu, Y. 2018. Recurrent neural networks for multivariate time series with missing values. *Scientific reports* 8(1): 1–12.

Chollet, F.; et al. 2018. Keras: The python deep learning library. *ascl* ascl–1806.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* .

Debosscher, J.; Sarro, L.; Aerts, C.; Cuypers, J.; Vandenbussche, B.; Garrido, R.; and Solano, E. 2007. Automated supervised classification of variable stars-I. Methodology. *Astronomy & astrophysics* 475(3): 1159–1183.

Goodricke, J. 1783. XXVI. A series of observations on, and a discovery of, the period of the variation of the light of the bright star in the head of medusa, called algol. In a letter from John Goodricke, Esq. to the Rev. Anthony Shepherd, DDFRS and Plumian Professor at Cambridge. *Philosophical Transactions of the Royal Society of London* (73): 474–482.

Hinners, T. A.; Tat, K.; and Thorp, R. 2018. Machine learning techniques for stellar light curve classification. *The Astronomical Journal* 156(1): 7.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* .

Jayasinghe, T.; Stanek, K. Z.; Kochanek, C. S.; Shappee, B. J.; Holoien, T. W. S.; Thompson, T. A.; Prieto, J. L.; Dong, S.; Pawlak, M.; Pejcha, O.; Shields, J. V.; Pojmanski, G.; Otero, S.; Hurst, N.; Britt, C. A.; and Will, D. 2019. The ASAS-SN catalogue of variable stars III: variables in the southern TESS continuous viewing zone. 485(1): 961–971. doi:10.1093/mnras/stz444.

Karpenka, N. V.; Feroz, F.; and Hobson, M. P. 2013. A simple and robust method for automated photometric classification of supernovae using neural networks. 429(2): 1278–1285. doi:10.1093/mnras/sts412.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Modak, S.; Chattopadhyay, T.; and Chattopadhyay, A. K. 2020. Unsupervised classification of eclipsing binary light curves through k-medoids clustering. *Journal of Applied Statistics* 47(2): 376–392.

Nair, V.; and Hinton, G. E. 2010. Rectified linear units improve restricted boltzmann machines. In *ICML*.

Neira, M.; Gómez, C.; Suárez-Pérez, J. F.; Gómez, D. A.; Reyes, J. P.; Hoyos, M. H.; Arbeláez, P.; and Forero-Romero, J. E. 2020. MANTRA: A Machine-learning Reference Light-curve Data Set for Astronomical Transient Event Recognition. *The Astrophysical Journal Supplement Series* 250(1): 11.

Sarro, L.; Sánchez-Fernández, C.; and Giménez, Á. 2006. Automatic classification of eclipsing binaries light curves using neural networks. *Astronomy & Astrophysics* 446(1): 395–402.

Shappee, B. J.; Prieto, J. L.; Grupe, D.; Kochanek, C. S.; Stanek, K. Z.; De Rosa, G.; Mathur, S.; Zu, Y.; Peterson, B. M.; Pogge, R. W.; Komossa, S.; Im, M.; Jencson, J.; Holoien, T. W. S.; Basu, U.; Beacom, J. F.; Szczygieł, D. M.; Brimacombe, J.; Adams, S.; Campillay, A.; Choi, C.; Contreras, C.; Dietrich, M.; Dubberley, M.; Elphick, M.; Foale, S.; Giustini, M.; Gonzalez, C.; Hawkins, E.; Howell, D. A.; Hsiao, E. Y.; Koss, M.; Leighly, K. M.; Morrell, N.; Mudd, D.; Mullins, D.; Nugent, J. M.; Parrent, J.; Phillips, M. M.; Pojmanski, G.; Rosing, W.; Ross, R.; Sand, D.; Terndrup, D. M.; Valenti, S.; Walker, Z.; and Yoon, Y. 2014. The Man behind the Curtain: X-Rays Drive the UV through NIR Variability in the 2013 Active Galactic Nucleus Outburst in NGC 2617. 788(1): 48. doi:10.1088/0004-637X/788/1/48.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems* 27: 3104–3112.

Vogel, H. C. 1890. Spectrographische Beobachtungen an Algol. *Astronomische Nachrichten* 123: 289.

Zechmeister, M.; and Kürster, M. 2009. The generalised Lomb-Scargle periodogram-a new formalism for the floating-mean and Keplerian periodograms. *Astronomy & Astrophysics* 496(2): 577–584.