CS 229 - Spring Quarter 2022

Classification of Variable Stars

Tirth Surti[1], Abhijit Devalapura[1], and Christina Sze[1]

[1]Stanford University, tsurti@stanford.edu, abhydev5@stanford.edu, csze@stanford.edu

July 3, 2022

1 Introduction

The study and classification of variable stars, stars whose brightness vary over time, is crucial for understanding the smaller time-scales of cosmic evolution and the distribution of stellar properties among different stars in the universe [1]. Furthermore, different types of variable stars serve as standard candles to measure distances to objects within and outside of the Milky Way.

Different types of stars have distinct variations in their brightness over time, allowing variable star classification to be a suitable task to apply machine learning models to. We focus on four main classifications of variable stars: Irregular, Eclipsing Binaries, Short-Period RR Lyrae, and Long-Period Mira, representative of the vast behaviors variable stars can and do exhibit. Irregular variables consist of Young Stellar Objects (YSOs) subject to irregular early stellar evolution events, red irregular variables, and rapidly rotating stars with irregular mass outflows [2]. Eclipsing Binary variables consist of a pair of stars that repeatedly overlap each other, causing periodic drops in brightness [2]. RR-Lyrae variables have very short orbital periods, ranging from several hours to a few days [3]. In contrast, Mira variables are red giant stars with periods that can range anywhere from one month to several years [2].

Variable stars are distinguished through the shape of their light curves, which plot the variation in the star's apparent magnitude (brightness) over time. Note that apparent magnitude of an object is its brightness observed from the Earth. These curves encode the intrinsic properties of star, like stellar activity, or extrinsic properties of a star, like being in a binary orbit with another object. We propose several different machine learning and deep learning methods that use these light curves to classify variable stars. We use standard machine learning algorithms, including softmax regression and neural networks that take in as input feature-extracted data from the light curves. We also use CNN-LSTM models that do not require any initial feature extraction and take in as input the sequential magnitude data.

2 Related Work

Traditional methods of analyzing and classifying variable stars involve the creation of phase-folded light curves, which find the best period for the data and shift all of the data into a single period of variation. These phase-folded light curves are analyzed for various patterns that can be used for classification [4]. Other traditional methods include the analysis of the features of light curves, including Fourier coefficients, periods, and amplitudes [5]. We aim to utilize some of these features as inputs to provide for an explainable and intuitive method for classification.

Early machine learning methods utilized such features that summarized the information from light curves. The earliest occurrence of this feature extraction approach was by Debosscher et al. (2007), who proposed 28 extracted features derived using Fourier analysis and harmonic fitting methods to use as input into various supervised learning techniques. However, some techniques, including support vector machines and Bayesian averaging of artificial neural networks, resulted in unsatisfactory classification accuracies [6]. Significant advances have since been made in feature extraction of light curves and the application of machine learning models towards variable star classification.

The random-forest classifier is the model most often used for variable star classification based off of feature extraction of light curves. Kim et al. (2014) extracted 22 features and used a random-forest model to categorize classes and sub-classes of variable stars, achieving a minimum of 80% accuracy on most variable star categories [7]. Com-

pared to these previous approaches, we believe that a more comprehensive set of features are necessary to boost performance on this classification task. We thus consider using more comprehensive and modern time series feature extractors.

Recently, there has been more research into the application of deep learning to the classification of variable stars. Aguirre et al. (2018) performed a study where they took the differences between the time and magnitude of light curves as their input and applied a 2D convolutional neural network [8]. This model found a significant increase in performance in classification compared to previous machine learning models. Bassi et al. (2021) proposed using a CNN-LSTM framework for variable star classification [4]. While such methods provide for minimal pre-processing and manual feature extraction, we believe that they are not fully explainable and do not utilize some of the more traditional techniques like phase-folding used by astronomers. We aim to improve upon these deep learning methods by integrating them with the more conventional phase-folding methods.

3 Dataset

Raw time-series magnitude data of different types of variable stars were obtained from ASAS-SN's online database, and it consists of time-series magnitude data for 687,462 variable stars [9][10]. A sample of raw light curves from each of the four categories is shown in Figure 1.

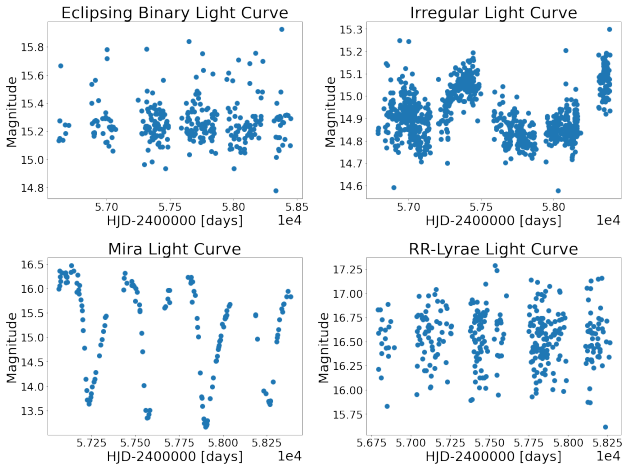

Figure 1: Sample Light Curves of Each Variable Star Type

Although it may appear that stars like eclipsing binaries and RR-Lyrae have an apparent periodicity on the order of hundreds to thousands of days given the regular gaps in the data, these variable stars typically have periods less than a day. So, there exists some finer variation in the magnitude of these stars that cannot be immediately seen from these plots. This finer variation is easier to see in phase-folded diagrams (see Figure 4). On the other hand, we can see the several-hundred-days-long variation in the brightness of the Mira variable star more clearly. We observe that the behavior of irregular variable stars does not follow any apparent pattern like that of the other star types.

For each of the four classification categories, we chose 5000 variable stars from the dataset. Given the modest dataset size chosen, we utilized a 60-20-20 split to obtain the training, development, and validation datasets so that enough examples that capture a sufficient amount of variation of the training dataset can be used for evaluation on unseen data. As raw light curve data cannot be used directly as input into classification algorithms, we conducted several methods to process and normalize the data. This is dependent on the model selection and will be explained in the next section.

4 Methods and Experiments

In our analysis, we worked with two different versions of the original dataset. We first implemented two basic models using a feature-extracted dataset from the raw time-series magnitude data. We then directly used the sequential magnitude and phase-folded data as input to the more sophisticated CNN-LSTM models in efforts to boost performance. Because this is a categorical classification task, all models are optimized based upon the cross-entropy loss function:

$$J(\theta) \equiv -\sum_{k=0}^{3} y_k \log \hat{y}_k. \tag{1}$$

4.1 Feature-Extracted Data Models

For each of the 20,000 stars, we kept up to 200 datapoints due to memory limitations. We then extracted 452 features using *tsfresh*, a comprehensive time-series feature extractor [11]. Sample features that were extracted include Fourier transform coefficients, partial autocorrelations, variances, peak counts, and absolute maximums, which are typical of some of the features traditionally used to analyze light curves.

To determine the significance of these features and see if there was any possibility for dimensionality reduction, we proceeded to perform Principal Component Analysis (PCA) on this feature-mapped dataset. To determine the number of components to keep, we used Minka's maximum likelihood estimation, which resulted in a total of 445 components, providing for little reduction in the

number of components [12]. Furthermore, when using just ten components for PCA, 58.2% of the variance of the data is explained, and we could only achieve 90% explained variance with 100 components. Given that *tsfresh* filters out irrelevant features that are unimportant or statistically insignificant, we thus chose to keep all 452 features.

4.1.1 Multi-Class Logistic Regression

As a baseline model, we first utilized a softmax regression model to see how well a linear classifer can distinguish between the different classes.

After splitting the dataset, we normalized the training data feature-wise and applied the same normalization transformation to the validation and testing datasets. We also integer-encoded the labels and then trained the linear classifer using ℓ_2 regularization, which adds a penalty $\frac{1}{2}\vec{w}^T\vec{w}$, where $\vec{w}$ is the weight vector, to the cross-entropy loss function. To find the parameters that minimize the loss, we used the limited-memory BFGS method, which is a quasi-Newton-Raphson method that approximates the inverse hessian at each step of the update to the weight parameters.

4.1.2 Fully-Connected Neural Network

A neural network can enable a more non-linear model to the dataset and reduce bias. We thus attempted to see whether a neural network can provide for better results using the same type of inputs. We found that a relatively shallow neural network was sufficient enough to quickly achieve lower bias than logistic regression in few iterations of batch gradient descent. However, to reduce overfitting, we applied ℓ_2 regularization on the first hidden layer with $\lambda = 0.01$, adding a penalty of $\lambda \sum_{i,j} W_{ij}^{[1]^2}$ to the cross-entropy loss function Equation 1. A summary of the model architecture is shown in Figure 2.

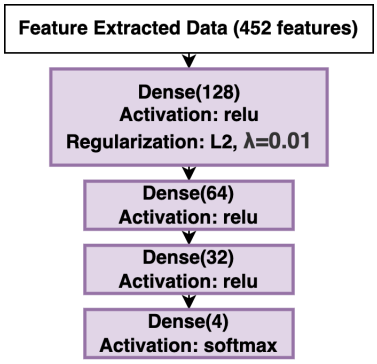

Figure 2: Diagram of Neural Network Used On Feature-Mapped Data

Similarly, we normalized the training data feature-wise and one-hot encoded the labels. Batch gradient descent was run with Adam optimization with a learning rate $lr_{fc} = 0.001$ for 20 epochs.

4.2 Models with Sequential Data

One of the main disadvantages of using feature extracted inputs is that significant computational resources are required to obtain relevant features. A CNN-LSTM model provides for less computational cost due to preliminary feature extraction as it can take as input directly the sequential data, which can be more efficient. Furthermore, this enables us to consider more datapoints for each star in the input, as we found increasing the upper limit of datapoints passed into *tsfresh* resulted in memory issues. Another key motivation for using CNN-LSTM models for variable star classification is that the magnitude data is inherently sequential over time, which can enable such models to extract temporal patterns and correlations over long and short time scales [4].

4.2.1 Single-Input CNN-LSTM Model

For the first CNN-LSTM model, we considered using the raw time series magnitude data for each star as direct input, which required minimal preprocessing. The magnitude of discrete points on each star's light curve was taken in order with time and a maximum of 500 datapoints were kept for each star. For stars with less than 500 datapoints, we zero-padded the sequences on the end so as to have the same size inputs.

We first applied two 64-filter and two 128-filter convolution layers and then a many-to-one bidirectional LSTM layer with 64 units to extract relevant temporal features. We used a bidirectional LSTM as both past and future datapoints are relevant for classification. These activations are then passed through a hidden layer to obtain classifications. We found that a single hidden layer was sufficient enough to achieve low bias.

Batch gradient descent with Adam optimization and a learning rate $lr_{si} = 0.001$ was done over 30 epochs. We limited the amount of epochs in training to reduce overfitting.

4.2.2 Triple-Input CNN-LSTM Model

To provide for a more explainable model that mimics traditional analysis of light curves, we introduce a CNN-LSTM model with multiple inputs. In addition to having as input the sequential time-sorted magnitude data as used in the first CNN-LSTM model, we also use as input the phase-folded light

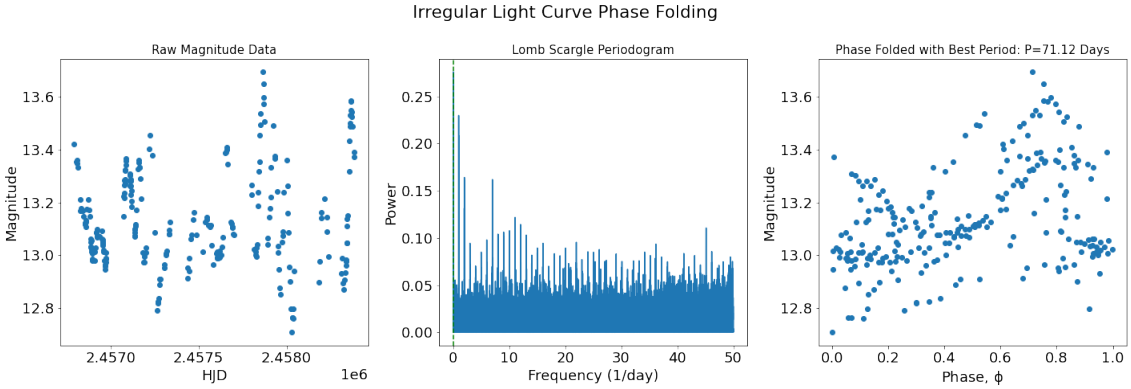

Figure 3: The creation of a cluttered phase-folded diagram (right) from an irregular variable star's light curve (left) using the best-period estimate from the Lomb-Scargle periodogram (middle).

curves for each variable star.

Phase folding squishes all of the magnitude data across multiple periods into a single period, allowing for a clearer representation of how magnitudes change over a period of variation. Given some reference time t_0, phases corresponding to each datapoint are calculated with:

$$\phi_n = \frac{t_n - t_0}{p} - \lfloor \frac{t_n - t_0}{p} \rfloor,$$

where $0 \leq \phi_n \leq 1$ indicates the n-th phase, t_n indicates the n-th time, p is the period, and $n \in \{x \in Z^+ | x \leq 500\}$ [13]. For consistency, we chose t_0 to correspond to the time with the lowest magnitude in the data. A sample phase curve for an RR-Lyrae variable star is shown in Figure 4.

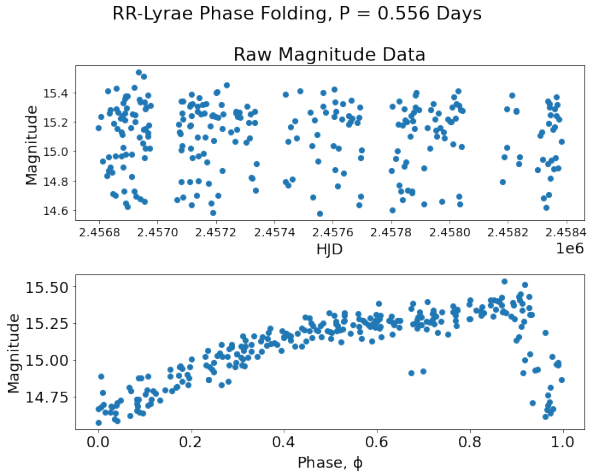

Figure 4: Phase Folding of RR-Lyrae Star

For irregular data, there is no known period, so a valid phase-folded diagram cannot be made. However, a cluttered phase diagram can still encode the irregularity of such star type. We used the Lomb-Scargle periodogram to estimate the best period by finding the frequency that corresponds to the highest power (see Figure 3) [14].

As a normalization technique, the difference between consecutive phases was obtained ($[\phi_1 - \phi_0, \phi_2 - \phi_1, \ldots \phi_{499} - \phi_{498}]$), where ϕ_t denotes the t-th phase), as well as the difference between consecutive light magnitudes ($[M'_1 - M'_0, M'_2 - M'_1, \ldots M'_{499} - M'_{498}]$, where M'_y denotes the magnitude of the variable star at the y-th phase). Along with the time-sorted magnitude data, these serve as the two additional inputs for the CNN-LSTM model.

For each input, we apply a series of convolutional layers, and for the sequential magnitude inputs, we also apply an LSTM layer for further time-series analysis. The activations are merged and flattened before being passed into hidden layers for classification. Figure 5 summarizes this model. We applied batch gradient descent with Adam optimization and a learning rate $lr_{ti} = 0.001$. We used ℓ_2 regularization with $\lambda = 0.001$ on the first hidden layer, adding a penalty of $\lambda \sum_{i,j} W_{ij}^{[1]^2}$ to the loss Equation 1, to reduce overfitting. Training was done only for ten epochs before no more improvement was seen on the validation dataset.

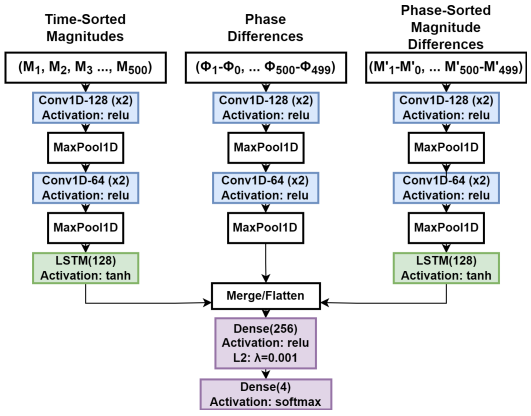

Figure 5: Triple-Input CNN-LSTM Architecture

	Precision				Recall				Average F1 Score
	EB	Irreg.	Mira	RR	EB	Irreg.	Mira	RR	
Softmax Regression	**0.846**	**0.907**	**0.993**	**0.890**	**0.884**	**0.904**	**0.992**	**0.852**	**0.908**
Fully Connected NN	**0.848**	**0.900**	**0.990**	**0.912**	**0.882**	**0.931**	**0.995**	**0.834**	**0.911**
Single Input CNN-LSTM	**0.899**	**0.931**	**0.992**	**0.912**	**0.938**	**0.902**	**0.985**	**0.907**	**0.933**
Triple Input CNN-LSTM	**0.972**	**0.966**	**0.984**	**0.958**	**0.968**	**0.975**	**0.990**	**0.948**	**0.970**

Table 1: Precision, Recall, and Average F1 Scores for All Models

5 Results and Discussion

As we seek to receive correct classifications for each star type and recover as many stars as possible from their appropriate classification, we chose to evaluate our models based upon precision and recall, respectively. We also reported the average F1 score (harmonic mean of the precision and recall) across all of the classes for each model so as to consider both recall and precision together and provide for direct model comparison. The results from each model are summarized in Table 1. Despite not having the highest precision and recall for every class (like Mira), we see that the Triple Input CNN-LSTM model achieves the best balance between precision and recall across all of the classes in general, achieving the highest average F1 score. This model's performance is summarized in Figure 6.

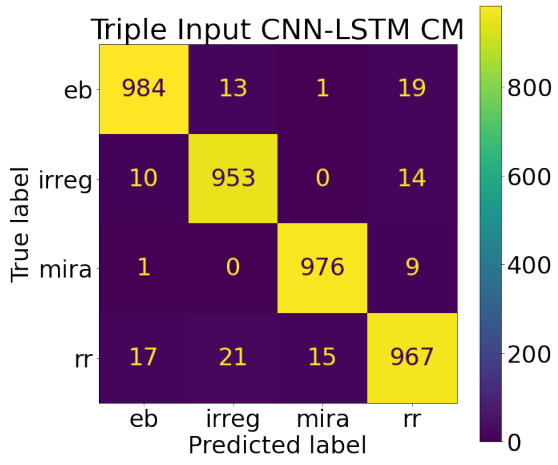

Figure 6: Confusion Matrix For Triple-Input CNN-LSTM

Comparing performance across the different input types, we see that we could achieve better performance using the sequential magnitude data in the CNN-LSTM models than the feature-extracted inputs from *tsfresh* in the basic softmax regression and fully-connected neural network models. This suggests that the CNN-LSTM models likely generated more significant features for classification than that of the other models. However, it is important to note that we allowed for a higher limit of the amount of datapoints (500 for the CNN-LSTM models and 200 for *tsfresh* due to memory limitations), which could correspond to better feature extraction in general.

Comparing performance within input types, we see that for the feature-extracted data, the fully-connected neural network performed only marginally better than the softmax regression model, suggesting that the feature-extracted data is not explained well by a significantly nonlinear model that a fully-connected neural network has the ability to produce. For the sequential data, we see that the triple-input CNN-LSTM model performs significantly better than the single-input CNN-LSTM model. This would make sense as including the additional phase data as input can further reveal distinguishing patterns between classes over a standardized domain, allowing for more relevant features to be extracted and analyzed.

6 Conclusion

The performance of the triple-input CNN-LSTM model provides for a novel, robust, and highly explainable method to boost classification performance on different types of variable stars. The acceptance of phase folded data as input into the triple-input CNN-LSTM model is an inherently human technique. Astronomers have long used phase-folding to classify variable stars, and using such input parallels these methods.

Future work on this project would likely involve analyzing and tuning model architectures on more granular variable star classifications. Within each of the four categories used, there exists more specific classifications, which provide for even subtler differences amongst light curves and hence a more difficult classification task.

Contributions

Tirth worked mainly on preprocessing the data to obtain features and the sequential/phase-folded data and implementing the CNN-LSTM models. Abhy and Christina worked on implementing the logistic regression and baseline neural network models and conducting PCA on the feature-extracted dataset. The text, images, and diagrams for the final write-up were implemented and created collectively. The GitHub repository containing the code used in this project is: Code for Variable Star Classification Project.

References

[1] Variables: What are they and why observe them? Last accessed 13 April 2022, https://www.aavso.org/variables-what-are-they-why-observe-them.

[2] The American Association of Variable Star Observers. Types of variable stars: A guide for beginners.

[3] Swinburne University of Technology. Cosmos: The sao encyclopedia of astronomy - rr lyrae.

[4] Saksham Bassi, Kaushal Sharma, and Atharva Gomekar. Classification of variable stars light curves using long short term memory network. *Frontiers in Astronomy and Space Sciences*, 8, 2021.

[5] M. Park, H.-S. Oh, and D. Kim. Classification of variable stars using thick-pen transform method. *Publications of the Astronomical Society of the Pacific*, 125(926):470–476, 2013.

[6] J. Debosscher et al. Automated supervised classification of variable stars. *Astronomy Astrophysics*, 475(3), 2007.

[7] D. Kim et al. The epoch project: I. periodic variable stars in the eros-2 lmc database. *Astronomy Astrophysics*, 566, 2014.

[8] C. Aguirre, K. Pichara, and I. Becker. Deep multi-survey classification of variable stars. *Monthly Notices of the Royal Astronomical Society*, 482:5078–5092, 2019. Last accessed 13 April 2022, https://academic.oup.com/mnras/article-pdf/482/4/5078/26899068/sty2836.pdf.

[9] T. Jayasinghe et al. The asas-sn catalogue of variable stars: I. the serendipitous survey. *Monthly Notices of the Royal Astronomical Society*, 477, 2018.

[10] B.J. Shappee et al. The man behind the curtain: X-rays drive the uv through nir variability in the 2013 active galactic nucleus outburst in ngc 2617. *The Astrophysical Journal*, 788, 2014.

[11] Julius Neuffer Andreas W. Kempa-Liehr Maximilian Christ, Nils Braun. tsfresh documentation, 2016-2021.

[12] Thomas P. Minka. Automatic choice of dimensionality for pca. *M.I.T. Media Laboratory Perceptual Computing Section Technical Report*, Nov 2000.

[13] Stefan Czesla, Sebastian Schröter, Christian P. Schneider, Klaus F. Huber, Fabian Pfeifer, Daniel T. Andreasen, and Mathias Zechmeister. PyA: Python astronomy-related packages, Jun 2019.

[14] Jacob T. VanderPlas. Understanding the lomb-scargle periodogram. *The American Astronomical Society*, 236(1), 2018.