J. Astrophys. Astr. (0000) **000**: ####
DOI

Learning to classify Variable Stars Light Curves using Long Short Term Memory Network

Saksham Bassi[1*], Kaushal Sharma[2], and Atharva Gomekar[3]

[1]Courant Institute of Mathematical Sciences, New York University
[2]Inter-University Centre for Astronomy and Astrophysics
[3]Georgia Institute of Technology
[*]Corresponding author. E-mail: sb7787@nyu.edu

Abstract. Owing to the current and upcoming extensive surveys studying the stellar variability, accurate and quicker methods are required for the astronomers to automate the classification of variable stars. The traditional approach of classification requires the calculation of the period of the observed light curve and assigning different variability patterns of phase folded light curves to different classes. However, applying these methods becomes difficult if the light curves are sparse or contain temporal gaps. Also, period finding algorithms start slowing down and become redundant in such scenarios. In this work, we present a new automated method, 1D CNN-LSTM, for classifying variable stars using a hybrid neural network of one-dimensional CNN and Long Short Term Memory (LSTM) network which employs the raw time-series data from the variable stars. We apply the network to classify the time-series data obtained from the OGLE and the CRTS survey. We report the best average accuracy of 85% and F1 score of 0.71 for classifying five classes from the OGLE survey. We simultaneously apply other existing classification methods to our dataset and compare the results.

Keywords. Deep Learning—Convolutional Neural Networks—Long Short Term Memory—Variable Star Classification.

1. Introduction

Variable stars have served a pivotal role in expanding our knowledge about various aspects of the universe. These systems have been extensively used for a vast range of studies with their implications on stellar and galactic astrophysics, cosmology, and planetary formation research. A few notable studies are estimating distances to galaxies within and beyond the Local Group and measuring the Hubble constant (Feast, 1999; Freedman et al., 2001; Clementini et al., 2003; Vilardell, Jordi & Ribas, 2007; Harris, Rejkuba & Harris, 2010; Riess et al., 2016; Bhardwaj et al., 2016; Ripepi et al., 2017), studying chemical composition of different galactic regions (Smith, 1995; Luck, Kovtyukh & Andrievsky, 2006; Pedicelli et al., 2009; Genovali et al., 2014), probing stellar structure and evolution (Catelan & Smith, 2015; Christensen-Dalsgaard, 2016; Das et al., 2020, and references therein), studying planetary formation through pre-main-sequence stars (Bell et al., 2013; Ribas, Bouy & Merín, 2015), etc.

Recent advancements in astronomical instrumentation has resulted in an avalanche of time-series data from dedicated time-domain surveys such as Optical Gravitational Lensing Experiment (OGLE; Udalski et al., 1993; Soszyński et al., 2015, 2016, 2018), All-Sky Automated Survey (ASAS; Pojmanski, 2002), Catalina Real-Time Transient Survey (CRTS; Drake et al., 2009; Djorgovski et al., 2011), Zwicky Transient Facility (ZTF; Bellm et al., 2019), and upcoming Vera C. Rubin Observatory (previously Large Synoptic Survey Telescope; LSST Science Collaboration et al., 2009), etc. These databases consist of multiple photometric observations with corresponding time-stamps (light curves) for different variable sources. Classifying these sources based on their light curves helps us in understanding the responsible mechanisms behind the variability and provides insight into their interior structure and formation. The abundance of data from the modern surveys and relevance of variable stars to various domains of astrophysics has heightened the need for automated methods for quick and accurate classification of variable star light curves.

Development of automated methods for classifying variable stars' light curves has seen an upward trend in recent years and has formed the core of many latest studies. A common approach for the automated classification is to extract the periodic and non-periodic features from the light curves and feed them to the machine-learning (ML) classifiers. Periodic features primarily consist of period and Fourier decomposition parameters whereas non-periodic features are mostly statistical parameters (Ferreira Lopes & Cross, 2017). An automated method developed by Debosscher et al. (2007) uses a set of 28 features which are derived from the Fourier analysis of the time-series. These features are mainly the amplitudes, phases, and frequencies obtained from the Fourier fit which are supplied to Gaussian Mixture and ML classifiers for the supervised training. Dubath et al. (2011) present an automatic classification process using statistical parameters such as mean, skewness, standard deviation, and kurtosis as classification attributes. They used Random Forest (RF) for the classification and also estimate the importance of each attribute. Richards et al. (2011) combine the periodic features with the non-periodic features proposed by Butler & Bloom (2011)). They demonstrate the application of various ML based classifiers to automatically classify large number of variability classes. They also attempt hierarchical classification using hierarchical single-label classification (HSC) and hierarchical multi-label classification (HMC) using RFs. Nun et al. (2015) offer a `Python` library called FATS (Feature Analysis for Time Series) intended to standardize the feature extraction process from a given time-series. Kim & Bailer-Jones (2016) develop a package, called UPSILoN, which uses 16 extracted features from the light curves and classifies light curves using Random Forest (RF) technique. Pashchenko, Sokolovsky & Gavras (2018) attempt the problem of variability detection using machine learning methods: Support Vector Machines (SVM), Neural Nets (NN), Random Forests (RF), etc. These methods are applied to 18 features which represent the scatter and/or correlation between points in a given light curve.

Feature based classification methods have shown to produce results with good accuracy but these methods make an inherent assumption about the availability of reasonable number of time-stamps for a given light curve. For example, UPSILoN recommends having more than 80 data points in a light curve for obtaining satisfactory precision and recall values. It is known that many light curves from these surveys are noisy and contain temporal gaps due to various reasons related to observational constraints and survey design. Also, the difference in cadence choices among different surveys can potentially make the feature-extraction (and classification) process heterogeneous and survey-dependent. For this reason the recent works have emphasized the need for classification models based on (i) the raw light curve data, or (ii) the features not requiring any pre-processing such as light curve folding, Fourier decomposition, etc. The feature extraction can be achieved either in a supervised or unsupervised fashion. These necessities are primarily driven by the fact that the time-series data might be sparse (and therefore not good enough to estimate the period) and can contain gaps in the observations.

Rather than providing the hand-crafted features for classification, recent studies focus on employing the raw time-series data and take advantages from the improved deep-learning (DL) frameworks. Mahabal et al. (2017) process the raw-light curves to generate *dm-dt* maps. These mappings reflect the difference between the magnitudes (dm) and the corresponding time-stamps (dt) for each pair in the light curve. These differences are binned in fixed dm and dt ranges to obtain the attributes having a uniform dimension for each light curve. These attributes are mapped in 2-dimensions as an image and corresponding class labels are provided to the Convolutional Neural Network (CNN) for training the model. Naul et al. (2018) demonstrate the use of recurrent neural network (RNN) based autoencoder for unsupervised and effective feature extraction. They use the latent space features at the end of encoding layers as representative of the light curves but with reduced dimensionality. This step addresses the issue of light curves with varying length which is a limitation with most ML/DL classifiers. However, they show that encoding-decoding process is more accurate with the period-folded light curves and therefore they use the latent space features obtained from the folded light curves for further classification using Random Forest. Aguirre, Pichara & Becker (2019) consider taking the difference between consecutive time and magnitude values in a light curve to generate two vectors, one each for time and magnitude. These difference vectors form a matrix with two rows and as many columns as the number of difference values. This matrix is passed to a 1D convolutional neural network for training the classification model where time and magnitude difference vectors are treated as two separate channels (similar to different color channels in an image).

In the present work, we use a deep-learning framework called Long Short-Term Memory (LSTM; Hochreiter & Schmidhuber, 1997) networks which are specifically designed for handling time-series data and propagate learning from the data to the deeper layers. These models are capable of learning long as well as short-term temporal features and can accommodate the input light curves of varying length. We propose a hybrid network of 1D CNN and LSTM model to classify light curves into different variability classes. Our approach is similar to the image captioning neural model introduced in Vinyals et al. (2014) which combined 2D CNN on images with the LSTM model. In the hybrid model, CNN layers learn to generate the features efficiently and the LSTM part carries out the task of finding the correlations among different observations at varying timescales of an input light curve. These are fed to to a fully connected classification layer to predict the variability class. For training the classification model, we use the raw light curve data as an input to the network without any feature-extraction process. This aspect gives this implementation an edge over the previous works. We also implement the 2D CNN method (Mahabal et al., 2017) on our dataset where we consider a modified strategy for generating the *dm-dt* images. We finally compare the results from the two approaches: 1D CNN-LSTM and 2D CNN.

This paper is organized as follows. In Sec. 2., we describe the data used in this work for training/testing the classification models and the pre-processing steps required for 2D CNN and 1D CNN-LSTM implementations. Following this, in Sec. 3. and Sec. 4., we briefly overview of the two deep learning techniques used for the classification. Sec. 5. presents the results obtained using the two approaches. Discussion on the results and conclusions are presented in Sec 6. with an outline of the road-map for the future work.

2. Data and Pre-processing

We obtain the archival time-series data from two surveys, OGLE and CRTS, for variability classification. Both datasets contain a large number of light curves and the respective variability classes (labels). We select only a few of these classes which have enough number of distinct light curves for a stable training of the classification model. We include five variable star classes from OGLE in our dataset, namely Classical Cepheids, δ Scuti, Eclipsing Binaries, Long Period Variables, and RR Lyrae. We consider only those light curves that have at least 100 data points to ensure that the network gets a sufficient number of data points to discover the correlations between them and extract useful features for classifying the type of variability. The OGLE database contains 165,105 light curves for Eclipsing Binaries but we select only 50,000 of them to avoid their dominance in the training and keep the computation feasible (though their number is still larger than the other four classes). The total number of OGLE light curves belonging to all the classes are 104,006 in our dataset. From the CRTS survey, we consider the light curves for seven variable star classes. These include Contact Binaries (EW), Detached Binaries (EA), three types of RR Lyrae (RRab, RRc, RRd), Rotating Variables (RSCVn) and Long Period Variables (LPV). The total number of CRTS light curves in our sample are 68,867. The summary of light curves selected in each variability class from the OGLE and CRTS surveys is provided in Table 1.

Table 1. Summary of light curves belonging to different variability classes obtained from the OGLE and CRTS surveys.

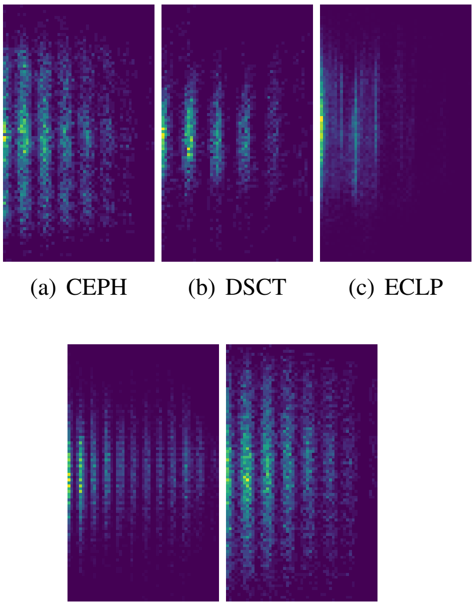
Class	Representation	Number
	OGLE dataset	
Classical Cepheids	CEPH	2698
δ Scuti	DSCT	464
Eclipsing Binaries	ECLP	50000
Long Period Variables	LPV	22371
RR Lyrae	RRL	28473
	CRTS dataset	
Contact Binaries	EW	30745
Long Period Variables	LPV	511
Detached Binaries	EA	4683
RR Lyrae type 1	RRab	2431
RR Lyrae type 2	RRc	28473
RR Lyrae type 3	RRd	502
Rotating Variables	RSCVn	1522

We notice that some variability classes are over represented in terms of the number of light curves in both datasets. However, we find that decreasing or increasing their number in the training sample just affects the training time and does not influence the classification accuracy in any significant manner. We use the two datasets separately for training and testing the 2D CNN and 1D CNN-LSTM models.

2.1 *Bi-dimensional histograms*

2D CNN model proposed in Mahabal et al. (2017) was applied to the light curves from the CRTS-North survey (CRTS-N; Mahabal et al., 2012). We apply the same method to a different dataset from the CRTS and OGLE surveys. As the name suggests, the 2D CNN model works on the two-dimensional datasets, like images, whereas light curves are uni-dimensional in nature depicting the variation in brightness as a function of time (Fig. 2(a)). Therefore we pre-process the light curves to generate bi-dimensional histograms (also called dm-dt mappings) and make them suitable for applying a 2D CNN model. To generate bi-dimensional histograms, we follow the same recipe as proposed in the source paper with a few modifications in the binning criteria as discussed below.

We compute the difference in magnitude and time for each pair of data points in the light curve. To find the optimal number of bins, rather than using fixed bin ranges, we used Freedman-Diaconis rule (Freedman & Diaconis, 1981). Freedman Diaconis estimator computes bin width using the number of data points in a sequence and interquartile range. The estimator was


(a) CEPH (b) DSCT (c) ECLP

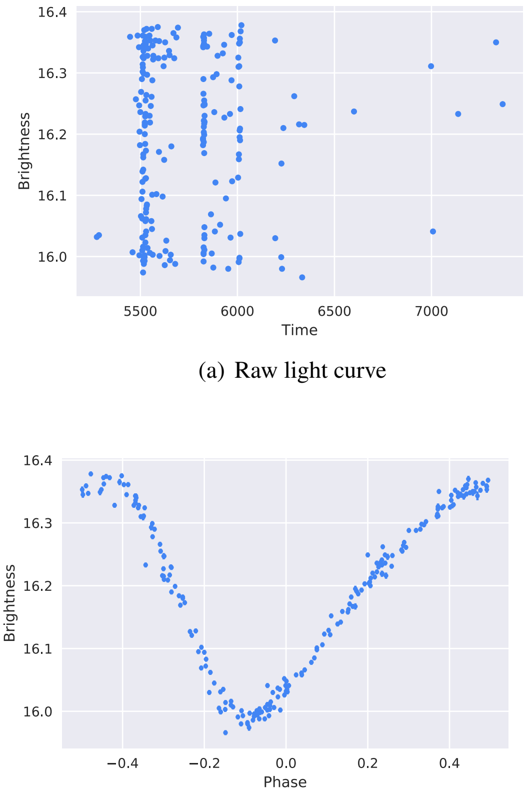
(d) LPV (e) RRL

Figure 1. Representative bi-dimensional histograms generated from the OGLE light curves for five variability classes.

applied to both variations, magnitude and time, individually. We adopt the Freedman-Diaconis rule as opposed to the original binning ranges as it takes into consideration that each light curve might have observations at different times. It automatically adjusts the bin range for a given density of points. We plot these bin sizes as histograms for checking the variations in magnitude and time. We finally consider the median of the maximum range in the histogram as the optimal number of bins for both time and magnitude variations. This provides the most frequent number of bins for a given survey. We find that an optimal number of bins for the CRTS survey for time and magnitude variations is 53 and 90 respectively. This results in an image of pixel size 90x53. We use the same number of time and magnitude bins for the OGLE survey. A few examples of bi-dimensional histograms generated from the OGLE light curves are shown in Fig 1.

2.2 *Padded time-series light curves*

The primary motivation to consider a 1D CNN model (with LSTM) comes from the fact that the raw light curves are one-dimensional sequences. Any rearrangement of these sequences can potentially cause information loss. However, if the variability is periodic in nature and the accurate period is known, the phase-folded light curve can be more informative than the raw light curve. Fig. 2 shows one example of a classical cepheid light curve from OGLE. It is clear that the raw light curve shown in the upper panel has no discernible pattern whereas the lower panel clearly shows a smooth sinusoidal-like variation with periodicity. Phase folded light curves have been used as an input to the classification model in various studies. But the process of period determination becomes challenging as well as computationally expensive in various circumstances such as lack of points in the light curves, insufficient coverage over different phases, etc. Therefore a classification model should be able to use the raw light curves without phase-folding or rearranging them.



(a) Raw light curve

(b) Phase-folded light curve

Figure 2. Scatter plots of a variable star belonging to Classical Cepheids from the OGLE survey. In the top panel, time on the X-axis is in MJD and the Y-axes in the two plots show the magnitude scale.

We propose a 1D CNN model that accepts the light curves without any pre-processing. The light curve data contains variable-length sequences but the computation of the classification model requires all input sequences of the same length. To make all light curves of the same length, we use zero padding at the end of the light curves. This step ensures that the shape of all input light curves remains consistent.

3. Convolutional Neural Networks

Convolutional Neural Networks (CNN or ConvNets; Hinton & Salakhutdinov, 2006; Bengio, 2009; Lecun, Bengio & Hinton, 2015) have been widely accepted as an excellent tool to identify patterns in astronomical data from diverse sub-domains (Dieleman, Willett & Dambre, 2015; Fabbro et al., 2018; Metcalf et al., 2019). These have been utilized in Astronomy for numerous classification and regression problems. Many studies have shown that a so-called deep-learning framework like CNN performs better than conventional machine learning algorithms (Kim & Brunner, 2017; Sharma et al., 2020).

A CNN is a kind of deep neural network commonly used for identifying features and patterns in imagery data. It consists of an input layer, an output layer, and multiple hidden layers. The hidden layers include convolutional layers, activation layers, pooling layers, fully connected layers, and normalization layers. Convolutional layers convolve the input image array with the filters and pass the outputs to the succeeding layer. Each convolution layer outputs a new set of activations. Pooling layers are used to decrease the size of the parameters by downsampling the features. Maxpooling is a type of pooling where the maximum value of a section of input is used in the next layer. This helps in generalizing the features and the reduction in features help CNN train quicker. Average pooling is another pooling technique where the average of the pooling window is used in the succeeding layer. Fully connected layers learn the features created from previous layers. Neurons in fully connected layers are connected to all activations of the previous layers. Normalization layers normalize the activations of the previous layers which helps in reducing overfitting. In this work, we use the same CNN architecture as prescribed in Mahabal et al. (2017). The schematic diagram of the architecture is shown in Fig. 3.

4. 1D CNN - LSTM

Inspired from the visual cortex system of animals (Hubel & Wiesel, 1968; Fukushima, 1980), CNNs are mainly used for learning local spatial features and the correlations among the neighbouring points in 2D signals like images. However, in recent years, the applications of CNNs have also been extended to other domains using one-dimensional data, e.g. spectral and time-series analysis, natural language processing, protein sequences, etc . Kiranyaz et al. (2019) provide a careful and detailed overview of 1D CNNs and discuss their applications to several problems. 1D CNNs have shown excellent capabilities to learn patterns and generating features from fixed length 1D data like time-series. We combine 1D CNN with the LSTM networks for classifying light curves.

Long Short Term Memory (LSTM) networks are a type of Recurrent Neural Networks (RNNs) having the ability to remember sequences for a long period of time. LSTM networks are used for learning from the sequential data like time-series, speech, and video (Graves & Schmidhuber, 2005; Wang & Jiang, 2015; Karim et al., 2018; Brunel et al., 2019). LSTM networks possess the property of selectively remembering past sections of the data. LSTM networks use three types of gates, namely update gate, forget gate, and output gate. The update gate is used in the current state. The forget gate is used to filter past outputs and the output gate is used to compute the final output. LSTM networks overcome the vanishing gradient problem by using the forget gate which helps in using previous outputs. This property helps LSTM networks to have an edge over conventional feed-forward neural network and Recurrent Neural Networks (RNNs). One example of LSTM networks applied to an Astronomical problem is provided in Zhang & Zou (2018). The authors implement the LSTM models for time-series prediction from the light curves. Evaluation of the results based on the mean squared error showed promising prediction for the future points in the time-series. A similar model has been shown to preform well in Czech, Mishra & Inggs (2018) for classifying transient radio frequency interference.

In this work, a hybrid network of 1D CNN and LSTM is used to classify the variable star light curves. LSTMs with 1D CNNs help the classification model to learn long and short term patterns, correlations, and dependencies in the input light curve. The convolutional layers in the network create features from the input sequences. They learn to create these features in a way to reduce the error in the prediction. These features are then used by the LSTM to produce its output activations. The output of the final LSTM layer is flattened. The flattened sequence of features is passed to fully connected layers that learn these features. The final softmax layer predicts the variable star classes with respective probabilities. The proposed networks consists of total 10 layers with 4 convolutional layers, 2 Maxpooling layers, 2 LSTM layers, and 2 fully connected layers. We use a kernel size of 3 for first 2 convolutional layers and 5 for last 2 convolutional layers. The filter size of convolutional and LSTM layers are in increasing fashion. It is followed by two fully connected layers containing 1024 and 512 neurons respectively. The final 'softmax' classification layer has five units for each of the five variability classes from OGLE.

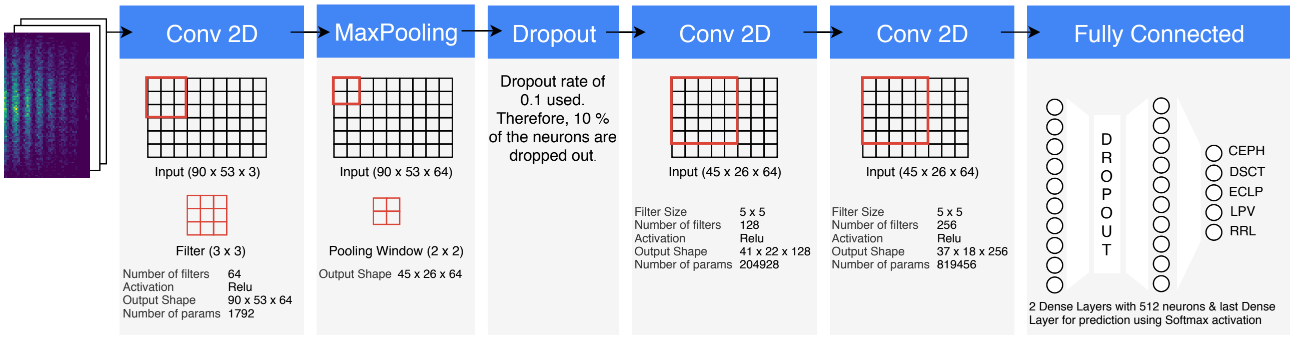


Figure 3. Schematic diagram showing the architecture of Convolutional Neural Network for the OGLE dataset. Different layers in the network are shown as different block and the layer names are indicated at the top of each block. The input images are of shape 90x53x3. First conv layer uses a kernel size of 3x3 but the subsequent layers uses a bigger kernel size of 5x5 to find bigger patterns. The same architecture is used for the CRTS dataset except the final classification layer which instead contains seven nodes corresponding to seven variablity classes present in the CRTS.

Full architecture of the model is presented in Fig. 4.

5. Results

For training the classification models described in Secs. 3. and 4., we use a single NVIDIA GeForce GTX 1060 6GB graphics processing unit. We use CUDA enabled Tensorflow (Abadi et al., 2015) environment for training the models in python. Since the number of light curves in different classes are highly imbalanced, therefore we give extra weights to the less presented class while training. The weights are assigned in accordance with the numbers of the light curves. To check the overall performance of the classification models, we use the standard metrics: precision, recall, accuracy, and F1 Score.

As described in Sec. 2., we need to generate the bi-dimensional histograms for providing as inputs to 2D CNNs. We use histogram2d function from python library NumPy to prepapre the Bi-dimensional histograms. We use 60%, 20%, and 20% of the ligthcurves for training, validation, and testing the classification models respectively. While dividing the dataset into training, testing, and validation samples, we ensure that the ratio of the variable classes remain the same across all the three data portions. We use Adam optimizer (Kingma & Ba, 2014) function with the learning rate fixed to 0.0002.

Application of 1D CNN-LSTM model does not require any pre-processing except the padding of the light curves. 1D CNN-LSTM has the advantage of using a smaller number of free parameters to perform classification. This could be a favourable approach in situations where the data is big and computing resources are limited. We use the same splitting criteria of 60%-20%-20% for the two datasets.

Table 2. Results of classification using 2D CNN on bi-dimensional histograms prepared from the OGLE and the CRTS survey light curves.

Dataset	Accuracy	Precision	Recall	F1 Score
OGLE	97.5%	0.81	0.91	0.85
CRTS	74.5%	0.56	0.52	0.54

5.1 *2D CNN on OGLE and CRTS*

We train and validate the 2D CNN model on 80% data from the OGLE and CRTS for multi-class classification problem where the number of classes are five and seven, respectively. The model is trained for 100 epochs and each training epoch takes about 165 seconds. Testing the CNN model on the remaining 20% data from OGLE gives an overall accuracy of 97.5%. For the CRTS dataset, we obtain an accuracy of 74.5%. As the number of light curves belonging to δ Scuti are on a lower side, we experiment with the classification using only the other four classes and get an overall accuracy of 99%. We note the similar increase in the accuracy for the CRTS dataset after removing classes with lesser number of light curves: LPV and RRd. The classification metrics using CNN for OGLE and CRTS are shown in Table 2. The normalized confusion matrix for OGLE is presented in Fig. 5).

5.2 *1D CNN-LSTM on OGLE and CRTS*

For applying 1D CNN-LSTM model, we examine the distribution of the number of observations in light

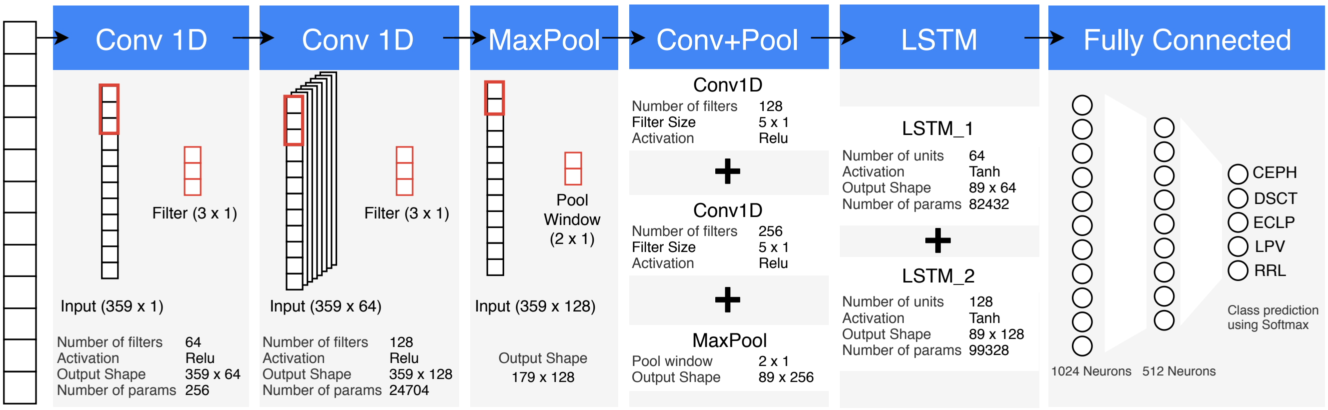


Figure 4. Schematic diagram showing the architecture of 1D CNN-LSTM model for the OGLE dataset. The input sequences are padded with zeroes to maintain the constant length of 359 in the OGLE survey and 546 in the CRTS survey for all light curves. Each block represent different type of layers as indicated by the header. The same architecture is used for the CRTS dataset except the final classification layer which instead contains seven nodes corresponding to seven variability classes present in the CRTS.

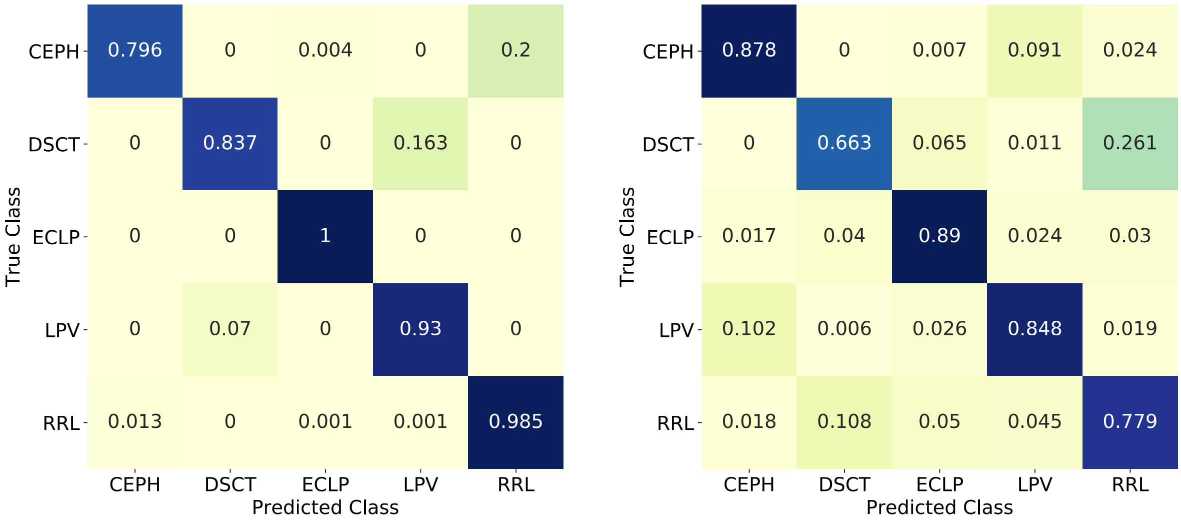


Figure 5. Classification results on the test set from OGLE using the 2D CNN (left) and 1D CNN-LSTM (right) models in the form of a normalized confusion matrix. Numbers in each cell represent the fraction of light curves belonging to a true and predicted class. Cells are color-coded according to the numbers in the cell.

Table 3. Results of classification using 1D CNN-LSTM on the OGLE and the CRTS survey light curves.

Dataset	Accuracy	Precision	Recall	F1 Score
OGLE	85.0%	0.64	0.81	0.71
CRTS	66.6%	0.46	0.53	0.49

curves in our dataset. We find that the distribution peaks at 359 for the OGLE and 546 for the CRTS. This means that most light curves in OGLE have 359 data points. To make all the light curves of equal length, the ones with lesser number of observation are padded with zeros at the end of the sequence. The light curves having more number of observations are clipped at 359th observation assuming that these observations will have sufficient information to classify the type of variability. We adopt the same strategy for the CRTS dataset.

We use the faster CuDNNLSTM layer which runs on GPU. The batch size of 32 was used for training 1D CNN-LSTM. The average time for one training epoch is around 75 seconds. Each classes is given weights according to the number of members in that class. We obtain average accuracy of 85% and 67% for the OGLE and the CRTS lightcurves, respectively. The results of 1D CNN-LSTM on both the datasets are presented in Table 3. The confusion matrix for the OGLE dataset is presented in Fig. 5.

6. Conclusions and Discussion

We present two approaches for classifying variable stars using Deep Learning techniques. While the 2D CNN model requires generating dm-dt mappings or bi-dimensional histograms, 1D CNN-LSTM does not require any pre-processing (except padding the light curves to maintain the uniform length for all the light curves) and is a step forward towards classifying light curves without providing engineered features or pre-processing. To classify bi-dimensional histogram images, we use the standard and well-established classification tool, the ConvNets, which perform very well on the OGLE dataset. We find that the classification performance on the CRTS dataset is suboptimal. In the second approach, we use a combined network of 1D CNN and LSTM to classify light curves. This approach does not require any pre-processing which saves a lot of time to generate bi-dimensional histograms. Also, the total training time using 1D CNN-LSTM is reduced by a factor of half as compared to the 2D CNN model. Despite gaining on the overall computation time front, we realize that the performance of 1D CNN-LSTM model is not at par with the 2D CNN approach. Lower accuracy hints that there could be potential shortcomings in the 1D CNN model which bars its performance as the feature extractor. It is also possible that the LSTM layers are unable to correlate the observations at varying time lengths. This aspect of disentangling the problem with the two types of layers needs further investigation.

The degraded performance on the CRTS dataset as compared to the OGLE dataset is a common difficulty faced by both the models. We note that ~86% of the light curves in the CRTS dataset come only from the two classes: the contact binaries (EWs) and RR Lyrae Type 2 (RRc). Since these two classes have very similar sinusoidal light curves with similar periodicity, most classification algorithms fail to distinguish between these two classes. It should also be noted that the CRTS dataset contains three RR Lyrae sub-classes: RRab, RRc, and RRd. RRab and RRc stars pulsate in the fundamental and first-overtone mode, respectively, whereas RRd stars pulsate in mixed mode between the fundamental and first overtone. The light curves for these classes also look very similar which makes the classification among these classes more difficult.

Despite giving a lesser accuracy, we are able to show that the proposed 1D CNN-LSTM model has the potential to perform the task of classifying variable star light curves without providing the processed data. Moreover, we see that 1D CNN-LSTM gives better results than the 2D CNN approach for certain classes in the CRTS dataset. For example, 1D CNN-LSTM performs better in separating out the light curves from EWs and RRc classes. For these two groups, 2D CNN is able to classify 42% RRc light curves correctly and labels wrongly 50% of the RRc light curves as contact binaries. On the other hand, 1D CNN-LSTM model shows marginally better accuracy of 45% for the RRc light curves and classifies incorrectly 41% light curves as contact binaries. About ~8% of the RRc light curves are classified as RRab. We make the same observation for the light curves from the RRab and EW classes. Only 29% of the RRab light curves are correctly classified by 2D CNN and 51% light curves are wrongly classified to EW class, whereas 1D CNN-LSTM classifies 55% of the light curves correctly and assigns wrongly only 22% of the light curves as EW. Similarly, while only 69% of the LPVs are correctly classified by the 2D CNN model, the 1D CNN-LSTM model achieves an accuracy of 80% in classifying them. These results, summarized in Table 4, highlight the superiority of the 1D LSTM-CNN model over 2D CNN in distinguishing very similar looking light curves.

As a future scope of this work, we would experiment more with the network architecture by optimizing the model hyperparameters, which requires more computational resources. We would also like to explore the capability of the hyperparameter optimized model

Table 4. Comparison of classification results for three variability classes from the CRTS dataset using 2D CNN and 1D CNN-LSTM models. The two rows for each true class represent the correct classification (to the actual class) and mis-classification(to the other class indicated in the second row) percentages, respectively.

True Class	Predicted Class	Classification models	
		2D CNN	1D CNN-LSTM
RRab	RRab	29%	55%
	EW	51%	22%
RRc	RRc	42%	45%
	EW	50%	41%
LPV	LPV	69%	80%
	RRab	13%	6%

in classifying light curves from different surveys and examining their performance in case of the sparse light curves. These tests would confirm the robustness of the model against various differences among surveys, e.g., cadence, instrumentation, etc. A more sophisticated approach to classify light curves could use a combination of two parallel CNNs, a 1D CNN for the light-curves and another 2D CNN for the science (or difference) images. The features generated using these two networks can be merged and transferred to the final classification layer. The 2D CNN in this alternative approach could provide the required assistance to the 1D CNN-LSTM model to surpass the achieved accuracy by a stand-alone 1D CNN-LSTM model.

Acknowledgements

We thank Ajit Kembhavi, Kaustubh Vaghamre, and Ashish Mahabal for their valuable suggestions and key inputs. KS acknowledges financial support from a Raja Ramanna Fellowship (10/1(16)/2016/RRF-R&D-II/630) awarded by the Department of Atomic Energy, Government of India.

References

Abadi M. et al., 2015, TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org

Aguirre C., Pichara K., Becker I., 2019, MNRAS, 482, 5078

Bell C. P. M., Naylor T., Mayne N. J., Jeffries R. D., Littlefair S. P., 2013, MNRAS, 434, 806

Bellm E. C. et al., 2019, PASP, 131, 018002

Bengio Y., 2009, Foundations and Trends in Machine Learning, 2, 1

Bhardwaj A., Kanbur S. M., Macri L. M., Singh H. P., Ngeow C.-C., Wagner-Kaiser R., Sarajedini A., 2016, AJ, 151, 88

Brunel A., Pasquet J., Pasquet J., Rodriguez N., Comby F., Fouchez D., Chaumont M., 2019, Electronic Imaging, 2019, 90

Butler N. R., Bloom J. S., 2011, AJ, 141, 93

Catelan M., Smith H. A., 2015, Pulsating Stars

Christensen-Dalsgaard J., 2016, arXiv e-prints, arXiv:1602.06838

Clementini G., Gratton R., Bragaglia A., Carretta E., Di Fabrizio L., Maio M., 2003, AJ, 125, 1309

Czech D., Mishra A., Inggs M., 2018, Astronomy and Computing, 25, 52

Das S., Kanbur S. M., Bellinger E. P., Bhardwaj A., Singh H. P., Meerdink B., Proietti N., Chalmers A., Jordan R., 2020, MNRAS, 493, 29

Debosscher J., Sarro L. M., Aerts C., Cuypers J., Vandenbussche B., Garrido R., Solano E., 2007, A&A, 475, 1159

Dieleman S., Willett K. W., Dambre J., 2015, Monthly Notices of the Royal Astronomical Society, 450, 1441

Djorgovski S. G., Donalek C., Mahabal A., Moghaddam B., Turmon M., Graham M., Drake A., Sharma N., Chen Y., 2011, arXiv e-prints, arXiv:1110.4655

Drake A. J. et al., 2009, ApJ, 696, 870

Dubath P. et al., 2011, MNRAS, 414, 2602

Fabbro S., Venn K. A., O'Briain T., Bialek S., Kielty C. L., Jahandar F., Monty S., 2018, MNRAS, 475, 2978

Feast M., 1999, PASP, 111, 775

Ferreira Lopes C. E., Cross N. J. G., 2017, A&A, 604, A121

Freedman D. A., Diaconis P., 1981, Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete, 57, 453

Freedman W. L. et al., 2001, ApJ, 553, 47

Fukushima K., 1980, Biological Cybernetics, 36, 193

Genovali K. et al., 2014, A&A, 566, A37

Graves A., Schmidhuber J., 2005, Neural Networks, 18, 602 , iJCNN 2005

Harris G. L. H., Rejkuba M., Harris W. E., 2010, Publ. Astron. Soc. Australia, 27, 457

Hinton G., Salakhutdinov R., 2006, 313, 504

Hochreiter S., Schmidhuber J., 1997, Neural Computation, 9, 1735

Hubel D. H., Wiesel T. N., 1968, Journal of Physiology (London), 195, 215

Karim F., Majumdar S., Darabi H., Chen S., 2018, IEEE Access, 6, 1662

Kim D.-W., Bailer-Jones C. A. L., 2016, A&A, 587, A18

Kim E. J., Brunner R. J., 2017, MNRAS, 464, 4463

Kingma D. P., Ba J., 2014, arXiv e-prints, arXiv:1412.6980

Kiranyaz S., Ince T., Abdeljaber O., Avci O., Gabbouj M., 2019, in ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8360–8364

Lecun Y., Bengio Y., Hinton G., 2015, Nature Cell Biology, 521, 436

LSST Science Collaboration et al., 2009, arXiv e-prints, arXiv:0912.0201

Luck R. E., Kovtyukh V. V., Andrievsky S. M., 2006, AJ, 132, 902

Mahabal A., Sheth K., Gieseke F., Pai A., Djorgovski S. G., Drake A., Graham M., the CSS/CRTS/PTF Collaboration, 2017, arXiv e-prints, arXiv:1709.06257

Mahabal A. A., Donalek C., Djorgovski S. G., Drake A. J., Graham M. J., Williams R., Chen Y., Moghaddam B., Turmon M., 2012, in IAU Symposium, Vol. 285, New Horizons in Time Domain Astronomy, Griffin E., Hanisch R., Seaman R., eds., pp. 355–357

Metcalf R. B. et al., 2019, A&A, 625, A119

Naul B., Bloom J. S., Pérez F., van der Walt S., 2018, Nature Astronomy, 2, 151

Nun I., Protopapas P., Sim B., Zhu M., Dave R., Castro N., Pichara K., 2015, arXiv e-prints, arXiv:1506.00010

Pashchenko I. N., Sokolovsky K. V., Gavras P., 2018, MNRAS, 475, 2326

Pedicelli S. et al., 2009, A&A, 504, 81

Pojmanski G., 2002, Acta Astron., 52, 397

Ribas Á., Bouy H., Merín B., 2015, A&A, 576, A52

Richards J. W., Starr D. L., Butler N. R., Bloom J. S., Brewer J. M., Crellin-Quick A., Higgins J., Kennedy R., Rischard M., 2011, ApJ, 733, 10

Riess A. G. et al., 2016, ApJ, 826, 56

Ripepi V. et al., 2017, MNRAS, 472, 808

Sharma K., Kembhavi A., Kembhavi A., Sivarani T., Abraham S., Vaghmare K., 2020, MNRAS, 491, 2280

Smith H. A., 1995, Cambridge Astrophysics Series, 27

Soszyński I. et al., 2015, Acta Astron., 65, 297

Soszyński I. et al., 2018, Acta Astron., 68, 89

Soszyński I. et al., 2016, Acta Astron., 66, 131

Udalski A., Szymanski M., Kaluzny J., Kubiak M., Krzeminski W., Mateo M., Preston G. W., Paczynski B., 1993, Acta Astron., 43, 289

Vilardell F., Jordi C., Ribas I., 2007, A&A, 473, 847

Vinyals O., Toshev A., Bengio S., Erhan D., 2014, arXiv e-prints, arXiv:1411.4555

Wang S., Jiang J., 2015, arXiv e-prints, arXiv:1512.08849

Zhang R., Zou Q., 2018, in Journal of Physics Conference Series, Vol. 1061, Journal of Physics Conference Series, p. 012012