

Classification of Variable Stars using Convolutional Neural Network

Abhina Premachandran Bindu
City College of New York, CUNY
New York, NY, United States
apremac000@citymail.cuny.edu

Greg Olmschenk
NASA Goddard Space Flight Center
Greenbelt, MD, United States
gregory.olmschenk@nasa.gov

Zhigang Zhu
City College of New York, CUNY
New York, NY, United States
zzhu@ccny.cuny.edu

Abstract—This study presents a deep learning-based framework for classifying variable stars using raw light curves from the Transiting Exoplanet Survey Satellite (TESS), an MIT-led NASA mission. As modern astronomical missions generate increasingly complex and high-volume time-series data, there is a growing need for scalable, automated classification systems. While prior efforts have applied machine learning to this domain, few have leveraged deep Convolutional Neural Networks (CNNs) directly on unprocessed TESS light curves. Our work addresses this gap by designing a CNN architecture capable of learning directly from the raw light curves, avoiding biases introduced by data transformations. The model classifies four key classes of variable stars - Delta Scuti, RR Lyrae, rotation modulation, and eclipsing binaries - and distinguishes them from non-variable or unknown cases. The classifier achieves high efficiency, processing each light curve in just tens of milliseconds on a single GPU, enabling large-scale inference. Applied to a set of 209,658 TESS light curves, the model identified 2,569 previously uncatalogued variable star candidates. The results demonstrate the potential of deep learning for accelerating discovery in astrophysical surveys and contribute to the broader intersection of data mining and physical sciences.

Index Terms—Machine learning, Convolutional Neural Networks, Variable Stars, Transiting Exoplanet Survey Satellite, Variable Star Catalog

I. INTRODUCTION

The primary objective of this study is to identify new variable stars and generate a catalog of variable star candidates by developing a 1D CNN that operates directly on unprocessed light curves from TESS.

A. Variable Stars

Variable stars are stars or star systems whose brightness changes over time. They play a crucial role in advancing our understanding of stellar evolution, stellar composition, exoplanet detection and habitability, intergalactic distance measurements, and more [3]. As such, research on variable stars contributes significantly to both particle physics and stellar physics. Variable stars are categorized on the basis of their underlying cause of variability. These stars are identified and distinguished based on characteristics such as period, shape and amplitude of their light curves [4]. There are mainly two types of variable stars: extrinsic and intrinsic. The variability of extrinsic stars is caused by changes in external properties such as rotation, whereas that of intrinsic stars is caused by changes in the physical properties of the star itself. In this work,

we are identifying and classifying four different classes of variable stars - Delta Scuti, RR Lyrae, rotation modulation, and eclipsing binaries. Delta Scuti and RR Lyrae are intrinsic and are pulsating variables, meaning their brightness variability is caused by the cyclical expansion and contraction of the star. These oscillations of the star's outer layers are caused by the imbalances between pressure and gravity [36]. Delta Scuti light curves show fluctuating amplitude and have periods ranging from approximately 0.01 to 0.2 days. On the other hand, RR Lyrae light curves are either sinusoidal or asymmetrical sawtooth shaped and have a longer period range from 0.2 to 1.2 days. The rotation modulation and eclipsing binaries variables are the extrinsic classes considered. The variability in rotation modulation objects considered are caused by solar like star spots or flares and their period can range from hours to years, depending on the rotation period of the star. As its name suggests, eclipsing binaries are binary star systems, and their variability is caused by one star eclipsing its companion. Its period can vary widely for each star as it depends on the orbital period of the stars in the system. A sample training light curve from each class is shown in Figure 1. These samples were selected to show the distinctive variable light curve features of each class. However, the features are often less prominent than those shown here.

B. TESS

The vast majority of stars in the galaxy are too far away to be spatially resolved into multiple pixels. Instead, one method of analyzing these stars is through photometry. That is, the brightness fluctuations of these stars are studied using light curves, which are the recorded magnitude or brightness of the stars over time. We are using light curves from the Transiting Exoplanet Survey Satellite (TESS) [37], an MIT-led NASA mission dedicated to observing millions of stars to detect transiting exoplanets - planets outside our solar system that periodically pass in front of their host stars, causing measurable changes in brightness [35]. Despite the vast extent of TESS light curves, which are well-suited for variable star study due to their ability to capture minor fluctuations in brightness, their use for variable star classification remains a relatively underexplored area of research.

Launched in 2018, TESS completed its two-year primary mission in July 2020 and is currently on its extended mission.

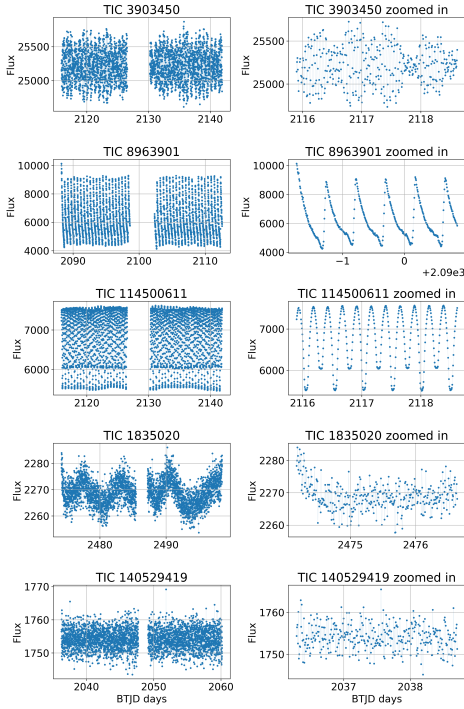


Fig. 1. Sample training light curves from each class are shown on the left. The same light curves zoomed on the time axis are shown on the right to display their characteristic shapes. The objects belong to Delta Scuti, RR Lyrae, eclipsing binary, and rotational modulation variable classes, and non-variable class respectively, in order from top to bottom. The TIC (TESS Input Catalog) ID of each object is displayed in the title of its respective light curve. Each light curve is plotted using the Barycentric TESS Julian Date (BTJD) on the X-axis and the PDCSAP (Pre-search Data Conditioning Simple Aperture Photometry) flux on the Y-axis.

The satellite is designed to scan the sky in 26 sectors during its primary mission, with half covering the Southern Hemisphere and the other half covering the Northern Hemisphere. TESS has observed or is scheduled to observe sectors 1 to 26 during the primary mission, sectors 27 to 55 during the first extended mission, sectors 56 to 107 during the later missions. Each sector covers a $96^\circ \times 24^\circ$ area of the sky and TESS is equipped with four wide-field $24^\circ \times 24^\circ$ cameras, each with four Charge-Coupled Devices (CCDs) [43] that enable high-quality photometric observations of each sector for 27.4 days. TESS primarily targets bright, nearby stars to facilitate the detection of exoplanets.

This research utilizes data collected during TESS’s first extended mission, specifically from sectors 27 to 55. During the extended mission, brightness measurements (flux) were recorded with a cadence of 2 minutes for target stars¹ and 10 minutes for full-frame images (FFIs). FFIs are the entire field of view of one of the four cameras and this is used to extract the light curves for the candidates not included in the target list [35]. The observations are captured using CCDs, and real-time onboard image processing is performed

¹Target stars are a list of stars selected for detailed observation by TESS. This includes stars that are considered good targets for transit detection [37].

by TESS’s Data Handling Unit to produce scientifically usable data products [15]. The data, collected as light curves in FITS (Flexible Image Transport System) files are stellar brightness variations or flux values recorded over time. Flux values used are pre-search data conditioning simple aperture photometry (PDCSAP) values and the time is measured in Barycentric TESS Julian Date (BTJD) units. Currently deployed on its extended mission, TESS continues to collect stellar data across different regions of the sky, offering ongoing opportunities for astrophysical research.

C. CNN Classifier

In order to classify the stellar light curves, one needs to inspect and deduce the period, amplitude and shape details of the light curve. And for hundreds of thousands of complex light curves, this is a tedious, repetitive, and error-prone task. We are automating this process by using a CNN, an artificial neural network (ANN) implemented using deep learning architecture. Deep neural networks excel in identifying complex patterns in complex data [10]. Among the various neural network frameworks, CNNs are widely recognized for their success in image classification and object detection tasks, largely due to their ability to identify hierarchical patterns and spatial correlations in data. While 2D CNNs are typically used for image-based data, our research focuses on one-dimensional sequential light curve data. Therefore, we utilize 1D CNNs to analyze the raw input directly. 1D CNN framework excels at efficiently extracting local patterns in 1D sequential data and have been used extensively in the natural language processing (NLP), diagnosis of medical sensors such as electrocardiograms (ECG) and electroencephalograms (EEG), audio signal analysis and financial data analysis [8]. Compared to Long Short-Term Memory (LSTM) networks [42], a type of Recurrent Neural Network (RNN) known for modeling temporal data, 1D CNNs offer greater computational efficiency. Although LSTMs are powerful for sequential data analysis and excel in modeling long-term temporal dependencies, they process data sequentially, typically making them slower than CNNs. In contrast, our 1D CNN classifier is highly efficient: on a single GPU, it can process a single light curve in just a few tens of milliseconds. Consequently, our tool enables the effective and efficient use of TESS data, supporting researchers in the exploration of variable star phenomena and contributing significantly to advancements in stellar studies.

II. BACKGROUND

A. From Perceptron to CNNs: Evolution of deep learning models

The very first Neural Network - the perceptron was introduced by F Rosenblatt in 1968 [5]. This simplest Neural Network framework was promising great results for simple linearly separable classification tasks. Later, in 1980, Kunihiko Fukushima proposed the first Neural Network for visual pattern recognition, named neocognitron [6]. Neocognitron was the very first predecessor for a CNN, which was inspired from the visual nervous system of vertebrates. Out of the various

types of neural networks emerged, CNNs have been proved to be exceptionally effective for computer vision tasks. CNNs have the special capability to deduce hierarchical patterns and spatial correlations in data using a kernel to generate feature maps from the input data. The very first CNN framework was introduced in 1989 by LeCun et al in the paper titled “Backpropagation Applied to Handwritten Zip Code Recognition” [9]. This paper demonstrates the effectiveness of backpropagation-based neural networks, specifically CNNs for the task of handwritten digit recognition on US zip codes from the USPS dataset. Later, in 2012, Krizhevsky et. al [11] proposed AlexNet, the first CNN that outperformed all other existing machine learning frameworks. AlexNet was the first CNN model to win the ImageNet challenge with an impressive least error rate and this became a turning point for the use of CNN frameworks for image classification and object detection tasks.

We are using a 1D multi-layered CNN, as this architecture is well-suited for capturing localized patterns in sequential data and offers improved computational efficiency through parallel data processing.

B. Related Research

Machine learning techniques were first applied to the study and classification of variable stars in the early 2000’s. One of the pioneering works in this area was by Debosscher et. al [12], who proposed the use of machine learning classifiers to analyze light curves from various astronomical missions, including HIPPARCOS (High Precision Parallax Collecting Satellite), OGLE (Optical Gravitational Lensing Experiment), and MOST (Microvariability and Oscillations of STars). They used a range of machine learning classifiers, such as a multivariate Gaussian mixture classifier, Bayesian networks classifier, Bayesian average of artificial neural networks and support vector classifier as models. This work focused more on developing a classifier optimized for speed, interpretability, and simplicity, rather than maximizing model performance.

Another relevant work by Aguirre et. al [14] published in November 2018, explores the classification of variable stars using a 1D CNN. However, the input data in this study are derived from three different surveys — OGLE-III, VISTA (Visible and Infrared Survey Telescope for Astronomy), and CoRoT (Convection, Rotation, and Planetary Transits) — and transforms the input data as a $2 \times N$ matrix, where 2 corresponds to the dimensions of time and magnitude values and N is the number of data points used for each light curve. Their CNN model includes only two convolutional layers, which lacks the capacity to learn intricate patterns the way deep learning methods do.

Later in 2020, Feinstein et. al [13] proposed a method on classifying short-cadence TESS light curves to identify stars with flares, analyzing flare statistics across different stellar age and spectral groups. Although stellar flares represent a type of stellar variability, this research emphasizes detecting localized spikes in the light curves rather than periodic variability. Their approach also utilizes TESS light curves and a CNN as the

classifier. However, the CNN architecture includes only two convolutional layers and a single dense layer, which may limit its ability to detect complex patterns in the data. In addition, the input light curves are preprocessed to remove long-term variability. Overall, while this study shares the use of TESS light curves and CNNs, its goals and methodology differ from ours.

The previous two models are both shallow networks. Deep neural networks, such as the one presented in this work, provide distinct benefits and challenges compared to their shallow counterparts.

Recently, in late 2024, Akhmetali et. al [2] proposed an approach that shares similarities with our research, as it employs a 1D CNN with multiple convolutional layers to classify variable stars. However, it differs significantly in methodology: the model inputs are folded versions of raw light curves, sourced from the OGLE dataset. They use this classifier to categorize known variable stars, and the use of folded light curves as input data may significantly affect the model’s efficiency.

Many types of light curve preprocessing transform the input data in ways that introduce biases or unwanted correlations, potentially leading to incorrect results. In contrast, our work distinguishes itself by using raw TESS light curves, leveraging a deeper CNN architecture with multiple convolutional layers, and avoiding preprocessing steps that may introduce bias.

III. DATA

A. Dataset Production

To train the CNN model, labeled light curves of known variable stars are essential. However, due to the absence of a comprehensive catalog of variable stars identified directly by TESS, training labels were sourced from external datasets, primarily from GAIA (Global Astrometric Interferometer for Astrophysics) mission’s archival data and catalogs published in research based on TESS observations. RR Lyrae and rotational modulation objects were identified from the GAIA archive [24], and their GAIA source IDs (unique identifiers for objects observed by the GAIA spacecraft) were retrieved. TIC IDs (unique identifiers for all objects considered from the TESS mission) for these objects were then obtained from Mikulski Archive for Space Telescopes (MAST), a NASA-funded repository hosting data from various space missions, by cross-matching² their GAIA source IDs with their corresponding TIC IDs using the TIC available from MAST. This was done using the Astroquery library [26], which provides a programmatic interface to query and access TESS data products through MAST. The TIC IDs of Delta Scuti objects were taken from the catalog published in a 2024 research paper by Olmschenk et. al [20], which focuses on using

²Cross-matching refers to the process of linking objects listed in different astronomical catalogs. In this study, GAIA source IDs were cross-matched with TIC IDs using the Astroquery interface, which queries the GAIA column available in the TIC. The values in GAIA column is the result of a precomputed positional cross-match [49] between the TIC and the GAIA catalog.

a CNN to identify short period variables from TESS data. Eclipsing binaries object TIC IDs were obtained from the catalog published in 2022 by Prša et. al[21], which includes 4,584 eclipsing binaries identified from TESS observations conducted during the mission’s primary phase.

TIC IDs for the negative class — i.e., objects with unknown variability, were obtained by excluding the TIC IDs associated with downloaded data for variable star classes from the complete list of TIC IDs available in MAST. Although this dataset likely contains some variable star light curves, we treat it as the negative class during training. This is justified because variable stars are comparatively rare, and thus the vast majority of these light curves are expected to represent non-variable stars. Consequently, using this dataset as a proxy for the negative class is a pragmatic and statistically sound approach for model training. For all the classes, the TIC IDs were split as 80%:10%:10% for training, validation and testing respectively. The total number of data for train, test and validation for each class is shown in table I.

Later, the astroquery tool is used for the process of downloading light curves for these obtained TIC IDs. The Science Processing Operations Center (SPOC) [19] at NASA Ames Research Center develops the calibrated pixels for the 2 min cadence data from TESS. The NASA Advanced Supercomputing (NAS) Division’s Pleiades supercomputer processes these data, including the full-frame image data, which are then archived in MAST. MAST facilitates the search and retrieval of TESS data through a web-based Archive User Interface [25]. While several access methods are available, this study utilizes the Astroquery.mast module [26] to retrieve the required TESS data.

Variable star class	train	validation	test
Delta Scuti	5781	704	659
RR Lyrae	379	54	50
Rotation modulations	2000	398	364
Eclipsing Binaries	6000	1086	1127
Unknown	99846	9999	99815

TABLE I
TABLE: COUNT OF TRAIN, VALIDATION AND TEST DATA USED FOR EACH CLASS

B. Data Cleaning

1) *Lomb-Scargle Periodogram for the folding of light curves*: The training data were manually reviewed through visual inspection to validate the classes as well as to find any discrepancies. Visually inspecting the period and amplitude shape of the light curves involves zooming in on the time axis to focus on a few cycles at a time. Given the extensive dataset, numbering in the hundreds of thousands, manually inspecting each light curve would be impractical and time-consuming. To address this, we used the Lomb-Scargle periodogram from the lightcurve package, a python library designed for analyzing time-series data from missions like TESS [28]. This tool automates the identification of the periods of these light curves and then folds them based on their periods (an example is shown in Figure 2), making the validation process easier

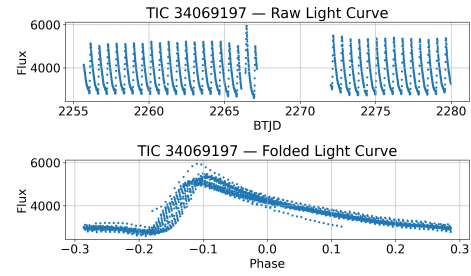


Fig. 2. Plot showing the raw and folded versions of light curve for an RR Lyrae with TIC ID 34069197

[27]. This automated folding of light curves worked well with objects with a definite period and amplitude, but failed at correctly finding the period of and folding a majority of objects in Delta Scuti and rotation modulation classes, as they have variable amplitudes. Nevertheless, the automation of light curve folding using Lomb-Scargle periodogram substantially aided in the validation of the accuracy of the training data.

2) *Removing overlaps*: Since it is natural for some stars to belong to multiple variable star classes simultaneously, overlap between classes is possible. For example, rotational modulation can occur in an eclipsing binaries system. Moreover, as each star may be observed in several TESS sectors, this results in multiple light curves for the same object. Consequently, even a small number of such overlaps can negatively affect model training. To mitigate this issue, the TIC IDs assigned to different classes were examined for overlap and duplicates were removed. A significant number of TIC IDs corresponding to eclipsing binaries were found to be present in other classes and were therefore excluded. Furthermore, before adding any new light curve to the training set, a check was performed to ensure that it was not part of the unknown dataset.

3) *Label cleaning*: Initially, this research considered six different classes of variable stars. In addition to the four primary classes mentioned earlier, Cepheids and short-timescale variables were also included. These objects were obtained by first collecting their source IDs from GAIA, as described in Section III-A. However, both classes were later removed due to issues related to data quality and the potential for overlap with other classes.

Upon reviewing the training light curves, it was found that a significant portion of the Cepheid data were corrupted, as illustrated in Figure 3. Specifically, the light curves showed trends that deviated significantly from the characteristic variability patterns of Cepheid stars. Even light curves from different sectors for the same object displayed inconsistent behavior, likely due to errors in data processing or issues during data collection in certain sectors. Alternatively, the inconsistencies may be due to potential issues with the GAIA labels.

Furthermore, while inspecting the dataset for overlapping and duplicate TIC IDs across different classes, a considerable number of shared objects were found between the short-timescale and eclipsing binaries classes. As the short-timescale

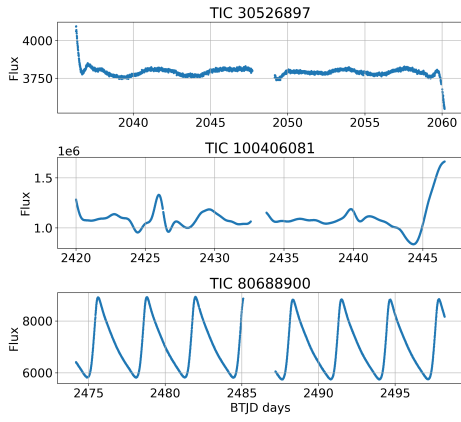


Fig. 3. Sample Cepheid light curves with discrepancies are shown, along with one without discrepancies for comparison. TIC IDs 100406081 and 30526897 represent objects with discrepant light curves, while TIC ID 80688900 corresponds to a Cepheid variable exhibiting a normal periodic trend.

class includes all variable stars with short periods, it is naturally more prone to overlap with classes such as Delta Scuti and short-period eclipsing binaries. Even in cases where TIC IDs did not directly overlap, the training light curves from the short-timescale class often exhibited trends similar to those found in other short-period variable classes. This similarity could lead to misclassifications during CNN training and negatively affect classification performance. Consequently, the short-timescale class was excluded from the final training set.

IV. METHODOLOGY

A. Convolutional Neural Network Architecture

Our neural network (NN) model, shown in Figure 4, is implemented using PyTorch’s [50] 1D CNN framework. Our model has eight convolutional layers and two dense (fully connected) layers and is based on the NN framework available at <https://github.com/golmschenk/ramjet> [39]. The architecture dynamically adapts to varying input lengths by computing appropriate pooling sizes and dense layer dimensions. By default, it expects an input length of 3500; longer light curves are truncated, while shorter ones are repeated to ensure a uniform input length across all samples. It comprises a sequence of convolutional and dense blocks and an end module tailored for multi-class classification. Each convolutional block comprises of a 1D convolutional layer followed by activation, pooling, and dropout layers. The size of the network increases progressively through the layers to the deeper layers to enhance network’s capacity to learn hierarchical features. That is, the number of neurons increases from 8 to 64 and then decreases as the network progresses towards the final dense layers, where the number of neurons is reduced to 20. Leaky Rectified Linear Unit (ReLU) [29] is the activation function used in all convolutional layers. Spatial dropout [30] regularization is implemented in the later layers to randomly drop a subset of feature maps during each training iteration.

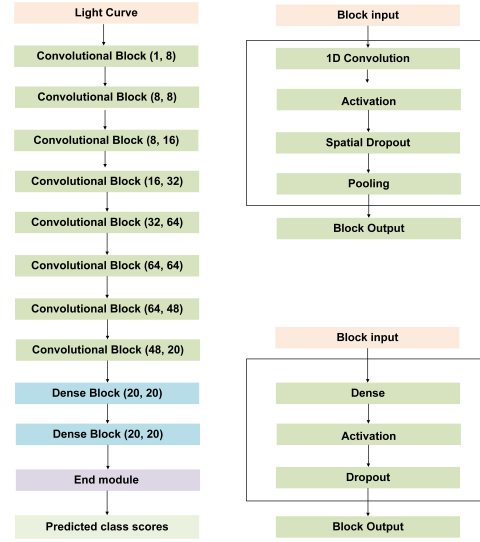


Fig. 4. NN architecture diagram illustrating the structure consisting of eight convolutional blocks followed by two dense or fully connected blocks. Each layer is annotated with a pair of values indicating the input and output dimensionality.

Data augmentation is implemented by creating new, synthetic training examples through injection and by applying a random roll, which creates a larger and more varied dataset. Together, these processes significantly mitigate the risk of overfitting. Overfitting refers to the tendency of an NN to fit the training data too closely, which impairs its ability to generalize and make accurate predictions on unseen data.

A pooling layer is applied after spatial dropout to reduce the spatial dimensionality of the feature maps [31]. Specifically, max pooling is used, which select the maximum value within a sliding window on the input. This operation reduces the input size while retaining the most significant features, effectively preserving essential information in the data. Finally, the output from all these layers is fed into the end module, which produces an unbounded score for each input. Although no activation function is applied within the end module itself, a softmax function is subsequently used during evaluation to convert these scores into class probabilities. The model is trained to classify inputs into one of five classes: four corresponding to types of variable stars, and one negative class representing objects not belonging to the other four. To address class imbalance—specifically, the under-representation of variable stars relative to non-variable stars—the training data was shuffled and resampled across multiple epochs. These combined strategies make our CNN framework well-suited for the efficient and accurate classification of one-dimensional temporal flux data from TESS.

V. EXPERIMENTAL EVALUATION

A. Training and Validation

The training pipeline is built on top of the qusi framework (<https://github.com/golmschenk/quasi>). The model and training code used is available at <https://github.com/Abhinagit24/>

variable_star_classification_CNN.git. Out of the total data prepared in section III for each class, 80% is designated for training and 10% for validation. The model's parameters are optimized during training using the Adam optimizer [40], with cross-entropy serving as the loss function.

Model training was carried out over multiple epochs, each consisting of training and validation steps with a batch size of 1000. About 100 training and evaluation runs were performed and for a single run, it took about 29 hours to complete 535 epochs. All experiments were conducted on a Linux-based workstation equipped with a 13th Gen Intel Core i9-13900KF processor (32 threads, 24 cores), 32 GB of RAM, and an NVIDIA GPU with CUDA support. The system was accessed remotely via SSH. The software environment consisted of Python 3.10, TensorFlow 2.12, and standard scientific libraries such as NumPy and Pandas. Training and inference tasks were executed locally without cloud-based resources or HPC clusters.

The learning rate was varied and evaluated across different training sessions. A higher learning rate of 0.01 led to unstable training behavior, characterized by sudden spikes and drops in accuracy. This instability is attributed to the large updates made to the model weights, which can overshoot optimal solutions and hinder convergence. Therefore, the learning rate was reduced to 0.001, which resulted in more stable training dynamics and improved learning, without such fluctuations.

In addition to tuning the learning rate, other hyperparameters were also adjusted to optimize model performance. These included the dropout rate, the size (number of neurons), and the depth (number of layers) of the neural network. The network size was varied by experimenting with different configurations, such as doubling or halving the number of neurons across layers. It was observed that reducing the number of neurons in the deeper layers led to improved accuracy.

The dropout rate was also tested across various values. A rate of 0.1 consistently produced the best results, effectively reducing overfitting while preserving model capacity. This dropout setting provided a good balance between regularization and learning ability. Beyond hyperparameter tuning, data quality and quantity were found to have a significant impact on performance. Removing corrupted and overlapping samples along with increasing the amount of training data, led to more substantial performance gains than hyperparameter adjustments alone.

The performance of the model was analyzed and compared using multiclass accuracy and multiclass area under the receiver operating characteristic curve (AUROC) metrics. A detailed explanation of these metrics is given in the following section. Our model achieved a training accuracy of 88.3% and a validation accuracy of 85.1%. Our network structure is shown in Figure 4.

1) *Performance metrics*: Even though our primary goal is to identify new variable stars and produce the catalog, a set of metrics was used to assess the model's performance and is primarily implemented through the PyTorch library. Since the data used is imbalanced, macro-averaging is applied, in

which each metric is computed separately for each class and then averaged. This approach helps ensure that all classes are weighted equally [46]. The metrics used to evaluate the model include cross-entropy loss, multiclass accuracy, and AUROC. These metrics are well suited for multi-class classification and allow evaluation of both the model's accuracy and its ability to distinguish between classes.

As this is a multi-class classification task, the loss function used is the categorical cross-entropy loss [1]. Its formula is given in equation 1.

$$L(\hat{y}, y) = - \sum_{i=1}^C y_i \log(\hat{y}_i) \quad (1)$$

where $L(\hat{y}, y)$ is the categorical cross-entropy loss, y_i is the true label, $\hat{y}_i$ is the predicted probability and C is the number of classes. The cross-entropy loss function focuses on the correct class predictions and maximizes the margin between correct class and remaining class predictions [45].

The model's performance on both the training and validation sets is evaluated using multi-class accuracy. Accuracy is the fraction value of correctly classified predictions over the total predictions [48]. It is represented by equation 2.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

where TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively.

For a classifier, the true positive rate (TPR) is the ratio of true positives to the total number of actual positives, while the false positive rate (FPR) is the ratio of false positives to the total number of actual negatives. The receiver operating characteristic (ROC) curve is generated by plotting the TPR against the FPR at various decision thresholds applied to the classifier [47]. In general, classifiers can be evaluated by comparing their ROC curves, with better-performing classifiers having curves that are higher and to the left in the ROC space [47]. The area under the ROC curve provides a single scalar value to assess classifier performance. It is particularly useful for comparing classifiers, as it offers a more reliable evaluation metric than accuracy, especially when dealing with imbalanced datasets, where accuracy may be misleading.

B. Test

Following the training and validation stages, a testing phase is conducted to assess the generalization performance of the classifier. The remaining 10% of the downloaded data was reserved for this purpose. The model's trained parameters were loaded from a previously saved checkpoint file in the form of a PyTorch tensor. The model is evaluated using the three primary metrics mentioned in section V-A1: cross-entropy loss, multiclass accuracy, and multiclass AUROC. During test phase, the model obtained a cross-entropy loss of 0.516, an accuracy of 85.9%, and an AUROC of 0.934. These results confirm that the trained CNN model generalizes effectively to new, unseen data.

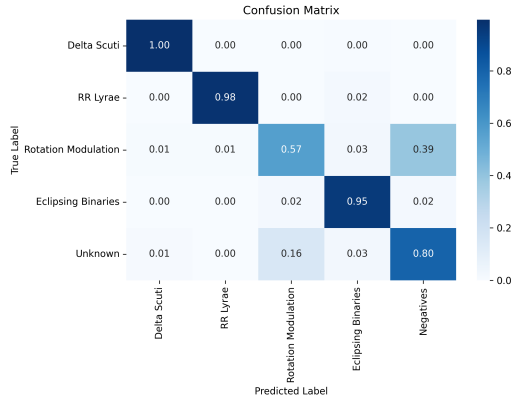


Fig. 5. Normalized confusion matrix

To evaluate the classification performance of the model by comparing actual versus predicted values, a confusion matrix was generated, with each row (representing the true label) normalized by the total number of samples in that row, as shown in Figure 5. A confusion matrix is a square matrix in which the rows represent the actual class labels and the columns represent the predicted class labels [46]. The diagonal elements represent correct classifications, while off-diagonal elements indicate misclassifications. The confusion matrix indicates that 39% of rotation modulation objects were misclassified as negatives, and 16% of unknown objects were misclassified as rotation modulations. These misclassifications are expected and likely stem from the presence of variable stars within the unknown class, which includes all objects with unconfirmed variability. Given that many stars exhibit flares or star spots, it is plausible that several rotation modulation sources were included in the unknown class, contributing to the observed misclassifications.

VI. CATALOG

A. Inference

After training and testing the CNN model, we performed inference to evaluate its performance and to identify new variable stars within a dataset of 209,658 light curves. These light curves, drawn from the entire "negative" class dataset, comprising the training, validation, and test splits—correspond to stars whose variability status has not been previously confirmed (see section III-A).

Testing the model on this mostly non-variable dataset is a strong way to measure how well it generalizes. If the model can correctly identify variable stars in data labeled as non-variable, it shows that the classifier is effective at finding new variable sources, even in uncertain or mislabeled data. During inference, the CNN model generated a confidence score for each light curve, from which the class with the highest score was assigned as the predicted label. Light curves with a confidence score greater than or equal to 0.95 were considered high-confidence predictions and selected for further analysis and inclusion in the final catalog (Table III).

To validate the model's predictions, 100 random samples from each predicted class were manually vetted by visual inspection. Of these samples, 98% of the Delta Scuti candidates, 96% of the RR Lyrae candidates, 98% of the rotation modulation candidates, and 99% of the eclipsing binaries candidates displayed light curve features consistent with their assigned classes, indicating strong reliability in the model's classification.

B. Catalog

The catalog of the newly found variables is given in the table III. It consists of newly found variable star objects with their TIC IDs, sectors, predicted class and confidence values. Objects with a confidence value of 0.95 or higher from the inference result are selected for the catalog. The classifier identified 4267 light curves, which belong to 2569 new objects, where 66 are RR Lyrae, 97 are rotation modulations, 820 are Delta Scutis and 1,586 are eclipsing binaries. This is listed in table II. As same objects were observed in different sectors, some TIC IDs are repeated across the catalog. Unique light curves can be identified by tracking TIC ID - sector pairs. The entire catalog is available at [51]

Variable star class	Number of objects found
Delta Scuti	820
RR Lyrae	64
Rotation modulation	97
Eclipsing Binaries	1586
Total	2569

TABLE II

TABLE: COUNT OF UNIQUE OBJECTS FOUND IN EACH VARIABLE STAR CLASS.

Some of the characteristic plots of the newly found objects are shown below. All the figures are colored by class such that, delta scuti, rr lyrae, eclipsing binaries and rotation modulation are colored as blue, orange, green and red respectively. Since eclipsing binaries typically consist of two stars, photometric measurements such as apparent magnitude and effective temperature represent the combined contributions of both components. As a result, the derived magnitude for an individual star is generally higher than that reported for the system as a whole, and the estimated temperatures may be less accurate due to the blending of the stellar fluxes. The normalized kernel density plot of TESS magnitude of each class of newly found objects were plotted and is shown in the Figure 6. The figure shows delta scuti stars as a slightly brighter class of objects, whereas rotation modulations appear as slightly dimmer ones. The normalized count plot of the log period values of newly found objects colored by class is shown in the figure 7. This plot is consistent with the known period distribution of the variable star classes considered, thereby validating the inferred classifications.

VII. CONCLUSIONS

Our classifier is both efficient and effective, successfully identifying new variable star candidates while processing each light curve in approximately 43 milliseconds on a GPU. It



Fig. 6. Plot showing normalized kernel density plots for magnitude of objects in each class.

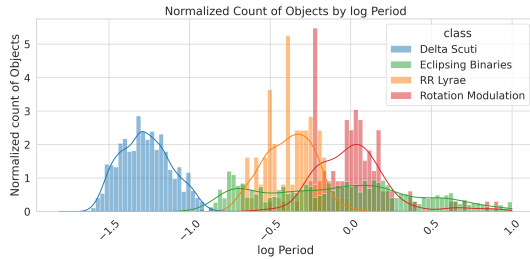


Fig. 7. Plot showing normalized count of period of newly found objects, colored by class.

detected 2,569 new variable star candidates among 209,658 light curves whose variability status was previously unknown. This result was generated by applying a confidence threshold of 0.95 to the inference results; only stars assigned a confidence of 0.95 or higher for a specific variable star class were included in the catalog under that class label. To assess classification quality, hundreds of these newly identified objects were randomly selected and visually inspected for periodic trends that were consistent with their assigned classes. The inspection showed that at least 96% appeared to belong to the correct predicted class.

The classifier achieved a training accuracy of 88.3%, a validation accuracy of 85.1%, and a test accuracy of 85.9%. On the test set, it also obtained a cross-entropy loss of 0.516 and an AUROC score of 0.934. The classifier efficiently generated a catalog of high-confidence, accurately predicted candidates using only unprocessed light curves, thereby fulfilling the primary objective of our study. In addition, this tool is designed to efficiently process large-scale time-series data, enabling automated classification and supporting further scientific investigation into stellar variability and its implications for stellar and particle physics.

Acknowledgments

This work has made use of data from the European Space Agency (ESA) mission *Gaia* (<https://www.cosmos.esa.int/gaia>), processed by the *Gaia* Data Processing and Analysis Consortium (DPAC, <https://www.cosmos.esa.int/web/gaia/>

[dpac/consortium](https://www.cosmos.esa.int/web/gaia/)). Funding for the DPAC has been provided by national institutions, in particular the institutions participating in the *Gaia* Multilateral Agreement.

This research also includes data collected by the TESS mission, which are publicly available from the Mikulski Archive for Space Telescopes (MAST). Funding for the TESS mission is provided by NASA's Science Mission directorate.

This work made use of Astropy:³ a community-developed core Python package and an ecosystem of tools and resources for astronomy.

REFERENCES

- [1] V. Acquaviva, Machine learning for physics and Astronomy. Princeton University Press, 2023.
- [2] A. Akhmetali et al., "Classification of Variable Star Light Curves with Convolutional Neural Network," *Galaxies*, vol. 12, no. 6, p. 75, Dec. 2024, doi: 10.3390/galaxies12060075.
- [3] L. Eyer et al., "GaiaData Release 2," *Astronomy & Astrophysics*, vol. 623, p. A110, Mar. 2019, doi: 10.1051/0004-6361/201833304.
- [4] M. Skarka and Z. Henzl, "Periodic variable A-F spectral type stars in the southern TESS continuous viewing zone," *Astronomy & Astrophysics*, vol. 688, p. A25, Jul. 2024, doi: 10.1051/0004-6361/202450711.
- [5] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958, doi: 10.1037/h0042519.
- [6] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, Apr. 1980, doi: 10.1007/BF00344251.
- [7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.
- [8] A. O. Ige and M. Sibiya, "State-of-the-Art in 1D Convolutional Neural Networks: A Survey," *IEEE Access*, vol. 12, pp. 144082–144105, 2024, doi: 10.1109/ACCESS.2024.3433513.
- [9] Y. LeCun et al., "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989, doi: 10.1162/neco.1989.1.4.541.
- [10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. Burges, L. Bottou, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf
- [12] J. Debosscher et al., "Automated supervised classification of variable stars: I. Methodology," *Astronomy & Astrophysics*, vol. 475, no. 3, pp. 1159–1183, Sep. 2007, doi: 10.1051/0004-6361:20077638.
- [13] A. D. Feinstein et al., "Flare Statistics for Young Stars from a Convolutional Neural Network Analysis of TESS Data," *The Astronomical Journal*, vol. 160, no. 5, p. 219, Oct. 2020, doi: 10.3847/1538-3881/abac0a.
- [14] C. Aguirre, K. Pichara, and I. Becker, "Deep multi-survey classification of variable stars," *Monthly Notices of the Royal Astronomical Society*, vol. 482, no. 4, pp. 5078–5092, Nov. 2018, doi: 10.1093/mnras/sty2836.
- [15] G. R. Ricker et al., "Transiting Exoplanet Survey Satellite (TESS)," *Journal of Astronomical Telescopes, Instruments, and Systems*, vol. 1, p. 014003, Jan. 2015, doi: 10.1117/1.JATIS.1.1.014003.
- [16] "TESS Archive Manual - TESS Archive Documentation Center - MAST," Accessed: Apr. 16, 2025. [Online]. Available: <https://outerspace.stsci.edu/display/TESS/TESS+Archive+Manual>
- [17] "Gaia Data Release 3 Documentation release 1.3," Accessed: Mar. 24, 2025. [Online]. Available: <https://gea.esac.esa.int/archive/documentation/GDR3/index.html>
- [18] T. Szklenár, A. Bódi, D. Tarczay-Nehéz, K. Vida, Gy. Mező, and R. Szabó, "Variable Star Classification with a Multiple-input Neural Network," *The Astrophysical Journal*, vol. 938, no. 1, p. 37, Oct. 2022, doi: 10.3847/1538-4357/ac8df3.

³<http://www.astropy.org>

- [19] J. M. Jenkins et al., “The TESS science processing operations center,” in *Software and Cyberinfrastructure for Astronomy IV*, G. Chiozzi and J. C. Guzman, Eds., in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 9913, Aug. 2016, p. 99133E. doi: 10.1117/12.2233418.
- [20] G. Olmschenk et al., “Short-period Variables in TESS Full-frame Image Light Curves Identified via Convolutional Neural Networks,” *The Astronomical Journal*, vol. 168, no. 2, p. 83, Jul. 2024, doi: 10.3847/1538-3881/ad55f1.
- [21] A. Prša et al., “TESS Eclipsing Binary Stars. I. Short-cadence Observations of 4584 Eclipsing Binaries in Sectors 1-26,” *The Astrophysical Journal Supplement Series*, vol. 258, no. 1, p. 16, Jan. 2022, doi: 10.3847/1538-4365/ac324a.
- [22] A. Ginsburg et al., “astroquery: An Astronomical Web-querying Package in Python,” *The Astronomical Journal*, vol. 157, no. 3, p. 98, Mar. 2019, doi: 10.3847/1538-3881/aafc33.
- [23] Gaia Collaboration et al., “The Gaia mission,” *ap*, vol. 595, p. A1, Nov. 2016, doi: 10.1051/0004-6361/201629272.
- [24] Gaia Collaboration et al., “Gaia Data Release 3. Summary of the content and survey properties,” *ap*, vol. 674, p. A1, Jun. 2023, doi: 10.1051/0004-6361/202243940.
- [25] D. Swade et al., “The TESS Science Data Archive,” in *Astronomical Data Analysis Software and Systems XXVII*, P. J. Teuben, M. W. Pound, B. A. Thomas, and E. M. Warner, Eds., in *Astronomical Society of the Pacific Conference Series*, vol. 523, Oct. 2019, p. 453.
- [26] M. St. Cyr, “Improvements to the accessibility and expansion of the astroquery.mast and astrocut tools,” in *American Astronomical Society Meeting Abstracts*, in *American Astronomical Society Meeting Abstracts*, vol. 243, Feb. 2024, p. 201.02.
- [27] J. T. VanderPlas, “Understanding the Lomb–Scargle Periodogram,” *The Astrophysical Journal Supplement Series*, vol. 236, no. 1, p. 16, May 2018, doi: 10.3847/1538-4365/aab766.
- [28] Lightkurve Collaboration et al., “Lightkurve: Kepler and TESS time series analysis in Python.” Dec. 2018.
- [29] X. Glorot, A. Bordes, and Y. Bengio, “Deep Sparse Rectifier Neural Networks,” in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, G. Gordon, D. Dunson, and M. Dudík, Eds., in *Proceedings of Machine Learning Research*, vol. 15, Fort Lauderdale, FL, USA: PMLR, Apr. 2011, pp. 315–323. [Online]. Available: <https://proceedings.mlr.press/v15/glorot11a.html>
- [30] A. Poernomo and D.-K. Kang, “Biased Dropout and Crossmap Dropout: Learning towards effective Dropout regularization in convolutional neural network,” *Neural networks: the official journal of the International Neural Network Society*, vol. 104, pp. 60–67, 2018, [Online]. Available: <https://api.semanticscholar.org/CorpusID:19245792>
- [31] K. O’Shea and R. Nash, “An Introduction to Convolutional Neural Networks,” 2015. [Online]. Available: <https://arxiv.org/abs/1511.08458>
- [32] V. Thakkar, S. Tewary, and C. Chakraborty, “Batch Normalization in Convolutional Neural Networks — A comparative study with CIFAR-10 data,” in *2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT)*, 2018, pp. 1–5. doi: 10.1109/EAIT.2018.8470438.
- [33] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014, [Online]. Available: <http://jmlr.org/papers/v15/srivastava14a.html>
- [34] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” in *Proceedings of the 32nd International Conference on Machine Learning*, F. Bach and D. Blei, Eds., in *Proceedings of Machine Learning Research*, vol. 37, Lille, France: PMLR, Jul. 2015, pp. 448–456. [Online]. Available: <https://proceedings.mlr.press/v37/ioffe15.html>
- [35] G. R. Ricker et al., “Transiting Exoplanet Survey Satellite,” *Journal of Astronomical Telescopes, Instruments, and Systems*, vol. 1, no. 1, p. 014003, Oct. 2014, doi: 10.1117/1.JATIS.1.1.014003.
- [36] D. M. Bowman, *Amplitude Modulation of Pulsation Modes in Delta Scuti Stars*. in *Springer Theses*. Cham: Springer International Publishing, 2017. doi: 10.1007/978-3-319-66649-5.
- [37] K. G. Stassun et al., “The TESS Input Catalog and Candidate Target List,” *The Astronomical Journal*, vol. 156, no. 3, p. 102, Sep. 2018, doi: 10.3847/1538-3881/aad050.
- [38] K. G. Stassun et al., “The Revised TESS Input Catalog and Candidate Target List,” *The Astronomical Journal*, vol. 158, no. 4, p. 138, Oct. 2019, doi: 10.3847/1538-3881/ab3467.
- [39] G. Olmschenk et al., “Identifying Planetary Transit Candidates in TESS Full-frame Image Light Curves via Convolutional Neural Networks,” *The Astronomical Journal*, vol. 161, no. 6, p. 273, May 2021, doi: 10.3847/1538-3881/abf4c6.
- [40] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization.” 2017. [Online]. Available: <https://arxiv.org/abs/1412.6980>
- [41] L. Biewald, “Experiment tracking with weights and biases,” *Software available from wandb.com*, vol. 2, p. 233, 2020.
- [42] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
- [43] M. Kunimoto, J. Winn, G. R. Ricker, and R. K. Vanderspek, “Predicting the Exoplanet Yield of the TESS Prime and Extended Missions through Years 1–7,” *The Astronomical Journal*, vol. 163, no. 6, p. 290, May 2022, doi: 10.3847/1538-3881/ac68e3.
- [44] R. J. Oelkers and K. G. Stassun, “Precision Light Curves from TESS Full-frame Images: A Different Imaging Approach,” *The Astronomical Journal*, vol. 156, no. 3, p. 132, Aug. 2018, doi: 10.3847/1538-3881/aad68e.
- [45] A. Demirkaya, J. Chen, and S. Oymak, “Exploring the Role of Loss Functions in Multiclass Classification,” in *2020 54th Annual Conference on Information Sciences and Systems (CISS)*, 2020, pp. 1–5. doi: 10.1109/CISS48834.2020.1570627167.
- [46] S. Raschka, Y. (Hayden) Liu, and V. Mirjalili, *Machine Learning with PyTorch and Scikit-Learn*. Packt Publishing Ltd., 2022.
- [47] C. E. Metz, “Basic principles of ROC analysis,” *Seminars in Nuclear Medicine*, vol. 8, no. 4, pp. 283–298, Oct. 1978, doi: 10.1016/S0001-2998(78)80014-2.
- [48] M. Grandini, E. Bagli, and G. Visani, “Metrics for Multi-Class Classification: an Overview.” 2020. [Online]. Available: <https://arxiv.org/abs/2008.05756>
- [49] P. Gavras et al., “Gaia Data Release 3. Cross-match of Gaia sources with variable objects from the literature,” *ap*, vol. 674, p. A22, Jun. 2023, doi: 10.1051/0004-6361/202244367.
- [50] A. Paszke et al., “Automatic differentiation in PyTorch,” in *NIPS-W*, 2017.
- [51] A. Premachandran Bindu and G. Olmschenk, “Classification of Variable Stars using Convolutional Neural Network Final Catalog”. Zenodo, Oct. 05, 2025. doi: 10.5281/zenodo.17272317.

VIII. APPENDIX

TABLE III: The catalog lists newly identified objects in each class, including their TIC ID, class, confidence, period, effective temperature, luminosity, TESS sector, GAIA G magnitude and TESS magnitude values. Only 25 of the objects are shown here.

TIC ID	Class	Confidence	Period	Teff	Lum	Sector	GAIAmag	Tmag
301289516	Eclipsing Binaries	0.97060847	0.2419	4199.0	0.112636611	42	9.83773	9.0684
371811873	Eclipsing Binaries	0.98248738	10.0	6882.0	24.6082745	38	11.8799	11.5162
157118534	Delta Scuti	0.98031288	0.0601	7168.0	9.355461	36	10.6272	10.3603
142197564	Eclipsing Binaries	0.96845698	0.7248	5089.0	0.263513	34	11.6913	11.1337
3664782	Delta Scuti	0.96001387	0.0365	7788.0	16.1757317	47	12.1194	11.9512
54039889	Eclipsing Binaries	0.98953724	2.3894	5782.0	3.41316223	36	12.0479	11.5745
386154220	Delta Scuti	0.98080027	0.0271	7738.86	9.774514	44	9.3648	9.21387
8007106	Delta Scuti	0.95857447	0.0592	7481.0	21.87407	38	10.6564	10.4367
58947120	RR Lyrae	0.98170853	0.3462	7669.74	13.5172729	45	10.1613	9.9777
235396448	Delta Scuti	0.95415151	0.0348	7370.0	8.871043	33	8.01348	7.81036
376860605	Delta Scuti	0.99166042	0.0696	7967.0	9.052658	55	11.0057	10.8573
428860793	Delta Scuti	0.991175771	0.0645	7466.0	9.866676	53	9.67501	9.42366
158960327	Delta Scuti	0.97960448	0.1245	6351.0	28.9438629	33	12.3316	11.4971
27846290	Delta Scuti	0.97552061	0.0556	7451.66	8.404558	41	11.5117	11.3027
219346938	Eclipsing Binaries	0.9821927	2.0204	6107.0	4.878466	33	11.4903	11.1046
3664782	Delta Scuti	0.98887604	0.0365	7788.0	16.1757317	44	12.1194	11.9512
19028616	Delta Scuti	0.96403229	0.0268			46	15.5152	15.3693
53997666	Delta Scuti	0.98366022	0.0453	8119.0	26.8373241	36	9.58683	9.4489
354407978	Eclipsing Binaries	0.96420991	0.2216	6763.0	5.18094349	55	10.8272	10.5257
318305863	Eclipsing Binaries	0.98239422	0.6417	6444.0	3.68667912	55	11.0239	10.68
148968781	Eclipsing Binaries	0.96718228	1.6973	5845.0	3.34725761	41	12.2393	11.8107
83461344	Delta Scuti	0.989631	0.0464	7527.99	16.9506035	44	9.83554	9.63025
373110694	Eclipsing Binaries	0.96580774	1.2144	6593.16	3.20259571	32	11.6798	11.2685
285149078	Delta Scuti	0.97452152	0.0524	9049.0	14.1966829	35	10.4757	10.3266
14879070	Eclipsing Binaries	0.96156067	0.1516	5284.47	0.852088153	47	13.1823	12.6389