Data Augmentation in a Hierarchical-Based Classification scheme for Variable Stars

Zafiirah Hosenie*
Jodrell Bank Centre for Astrophysics
Department of Physics and Astronomy
The University of Manchester, Manchester M13 9PL, UK.
zafiirah.hosenie@gmail.com

Robert Lyon
Department of Computer Science
Edge Hill University
Ormskirk Lancashire L39 4QP, UK
lyonro@edgehill.ac.uk

Benjamin Stappers
Jodrell Bank Centre for Astrophysics
Department of Physics and Astronomy
The University of Manchester, Manchester M13 9PL, UK
Ben.Stappers@manchester.ac.uk

Arrykrishna Mootoovaloo
Department of Astrophysics
Imperial College London
Prince Consort Road, SW7 2AZ
a.mootoovaloo17@imperial.ac.uk

Vanessa McBride
Department of Astronomy
University of Cape Town
Private Bag X3 Rondebosch 7701, South Africa
vanessa@astro4dev.org

Abstract

The accurate automated classification of variable stars into their respective sub-types is difficult. Machine learning based solutions often fall foul of the imbalanced learning problem, which causes poor generalisation performance in practice, especially on rare variable star sub-types. We attempted to overcome such deficiencies via the development of a hierarchical machine learning classifier. This 'algorithm-level' approach to tackling imbalance, yielded promising results on Catalina Real-Time Survey (CRTS) data. We attempt to further improve hierarchical classification performance by applying 'data-level' approaches to directly augment the training data so that they better describe under-represented classes. We apply and report results for three data augmentation methods in particular: *R*andomly *A*ugmented *S*ampled *L*ight curves from magnitude *E*rror (`RASLE`), augmenting light curves with Gaussian Process modelling (`GpFit`) and the Synthetic Minority Over-sampling Technique (`SMOTE`). When combining the 'algorithm-level' (i.e. the hierarchical scheme) together with the 'data-level' approach, we further improve variable star classification accuracy by 1-4%. We found that a higher classification rate is obtained when using `GpFit` in the hierarchical model.

1 INTRODUCTION

A major issue that impedes the successful automated classification of astronomical data is the imbalanced learning problem. This problem impacts classification of variable stars in particular, as some types of variable stars are rare or unusual phenomena, making it difficult to build automated machine learning (ML) systems to be able to recognize them. In previous work, we attempted to

*https://zafiirah13.github.io/zafiirah-hosenie/

Third Workshop on Machine Learning and the Physical Sciences (NeurIPS 2020), Vancouver, Canada.

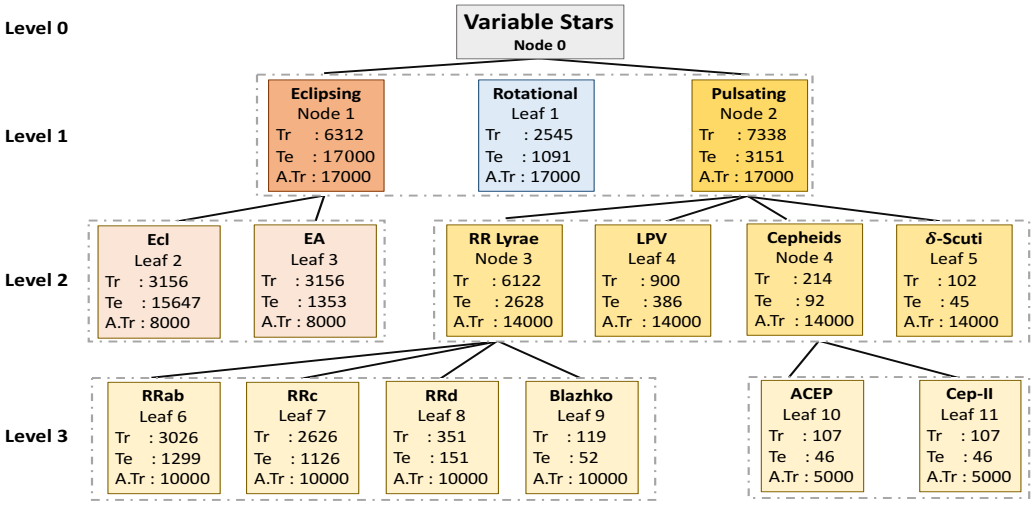

Figure 1: Hierarchical Tree classification with automated light curves augmentation for CRTS Data. The number of training examples (real LCs) is represented by *Tr*, the number of training examples after augmentation (both real and synthetic LCs) is represented by *A.Tr* and the number of test examples (real LCs) is represented by *Te*. At level 1, the real LCs in the training set are augmented and the dotted square represents a trained model (RF/XGBoost classifier). During testing phase, the classified examples in the test set move down the hierarchy at level 2. Afterwards, real LCs in the training set in level 1 moves to their respective branches at level 2. The real LCs are augmented and features are extracted. This process is repeated until it reaches all leaves in the hierarchy.

develop a variable star classifier together with various feature selection and importance techniques, and ran into the imbalanced learning problem. To overcome this, we attempted to modify the algorithms used for classification, and ultimately proposed a successful hierarchical classification system. Whilst the hierarchical system was effective, recall on minority classes could be stubbornly low relative to majority classes. In other domains, such problems are overcome by balancing the training distribution directly.

Therefore, this paper is mostly concerned with learning from an imbalanced class distribution. The problem is typically addressed using the following approaches: (i) *data level*: We employ three data augmentation approaches in conjunction with the hierarchical classification (HC) scheme for variable stars (see Fig. 1) in such a way that the class distributions are rebalanced directly; that is, it is a first proof of principle for the application of a level-wise augmentation within Hierarchical taxonomy for this problem, where we resample the original data set to achieve a desired balancing. (ii) *algorithm level*: We focus on using two different algorithms Random Forest (RF; [1]) classifier and eXtreme Gradient Boosting (XGBoost; [2]), together with a Bayesian optimization algorithm for hyperparameter tuning, to achieve improved performance on the minority class examples.

2 DATA

The Catalina Real-Time Transient Survey (CRTS; [3]) has produced a catalogue of periodic variable stars from 6 years of optical photometry. Each variable star is described by its time, magnitude and error in magnitude, also known as light curve (LC). We consider only 11 classes of variable stars which are heavily imbalanced as illustrated in Table 1. Thus to simplify our experimentation, we reduced the size of the largest class (Ecl) via random undersampling. We downsample this class to 4509 (this makes the number of Ecl examples comparable to the next biggest class, EA) and the remaining Ecl light curves (LCs) are then used for prediction.

2.1 DATA AUGMENTATION TECHNIQUES

While the undersampling methods (i.e. downsample Ecl and developing the hierarchical system) help to address some of the class imbalance issues, they are themselves insufficient, as minority class

performance was not good enough for our purposes. We therefore provide three ways to oversample the data, a form of data augmentation, which is necessary as some of the classes still outnumber other classes (see Tr values in Fig. 1). We augment the data via the generation of artificial data in order to increase the number of training samples by generating similar but not identical examples. In this work, we consider three methods of augmentation: (i) `SMOTE`, (ii) `RASLE`, and (iii) `GpFit`.

Synthetic Minority Oversampling Technique: `SMOTE` inserts artificially generated minority class examples into a data set by operating in 'feature space' rather than 'data space'. This technique helps to balance the overall class distribution.

***R*andomly *A*ugmented *S*ampled *L*ight curves from magnitude *E*rrors**: `RASLE` is employed on LCs or time series data directly. Using this information, we generate new light curves in the following way. Let us consider a probability distribution which can be concisely represented by a normal distribution. The probability distribution function (*pdf*) can be interpreted as going over the magnitude space vertically with the horizontal axis showing the probability that some value will occur. To construct the *pdf*, we make an assumption that the magnitude follows a normal distribution with mean, μ, to be equal to the true magnitude and the standard deviation, σ, to be equal to the error in magnitude. For each data point at a specific time, we sample a single magnitude from the *pdf*. Each sampled magnitude is assigned the same time as in the original data. The generated light curve is given the new (random) sampled magnitude with the same time value as in the original data. An illustration of `RASLE` is shown in Fig 3.

Modelling light curves with Gaussian Processes: An ideal case for data augmentation is to use a well-defined model of the classes under consideration to create synthetic data. However, there is no available model valid for all the different variable stars considered. We therefore build a model describing variable stars using Gaussian Processes (GPs, [6]) applied to CRTS data. In our case, we want to model light curves, so we require a kernel for GP that can demonstrate both small fluctuations and smooth variations. Given the different characteristics of the various stars, an appropriate choice of the kernel in this work is the Matern 5/2 kernel. Using the Limited memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS; [4]) optimization algorithm, the kernel hyperparameters are optimised and are used to fit the GP from which we sample synthetic light curves to augment our training set. Before fitting a GP to our data, we first convert the LCs from time distribution to phase distribution (folded-light curves) where the data is at the detected period for each variable star. We then randomly sampled synthetic LCs from the GP model to form the augmented training set. We show an example of `GpFit` on the folded-LCs for the different variable stars in Fig. 2 and the bottom plot illustrates 3 synthetic LCs randomly drawn from `GpFit`. We then unfolded the phases back into time space and used those synthetic LCs together with the original LCs as the training set.

3 METHOD DESCRIPTION

We provide an overview of the 'data-level' HC scheme, together with various stages we adopt to build the ML pipeline.

Stage 1: hierarchical tree classifiers - We use the astrophysical properties of the various sources to construct a hierarchical-based structure to represent the different classes (Fig. 1). Each node/leaf represents a class – identified by the label inside the node/leaf – and the edges represent the relationship between the superclass and subclass. For the HC, we use XGBoost and RF and then report the one that provides the best classification performance.

Stage 2: level-wise data augmentation in HC - Since the training set is still imbalanced after aggregating subclasses into superclasses, we use the three data augmentation techniques. Each technique is applied and tested independently in our HC based ML pipeline. For the `SMOTE` approach, features (the mean magnitude, standard deviation, skewness, kurtosis, mean-variance, amplitude and period) are extracted from the real LCs. Then, `SMOTE` automatically balances the class distribution via the creation of synthetic examples sampled over the feature space, such that the size of the minority class equals the size of the majority class, cancelling the imbalance out. This process is repeated for each branch and level in the HC, where the training set is directly balanced according to the size of the majority class prior to data augmentation.

While for the `GpFit` and `RASLE` cases, we are generating new light curves based on real LCs, thus generating new synthetic LC examples. Therefore, our training set will consist of both real and

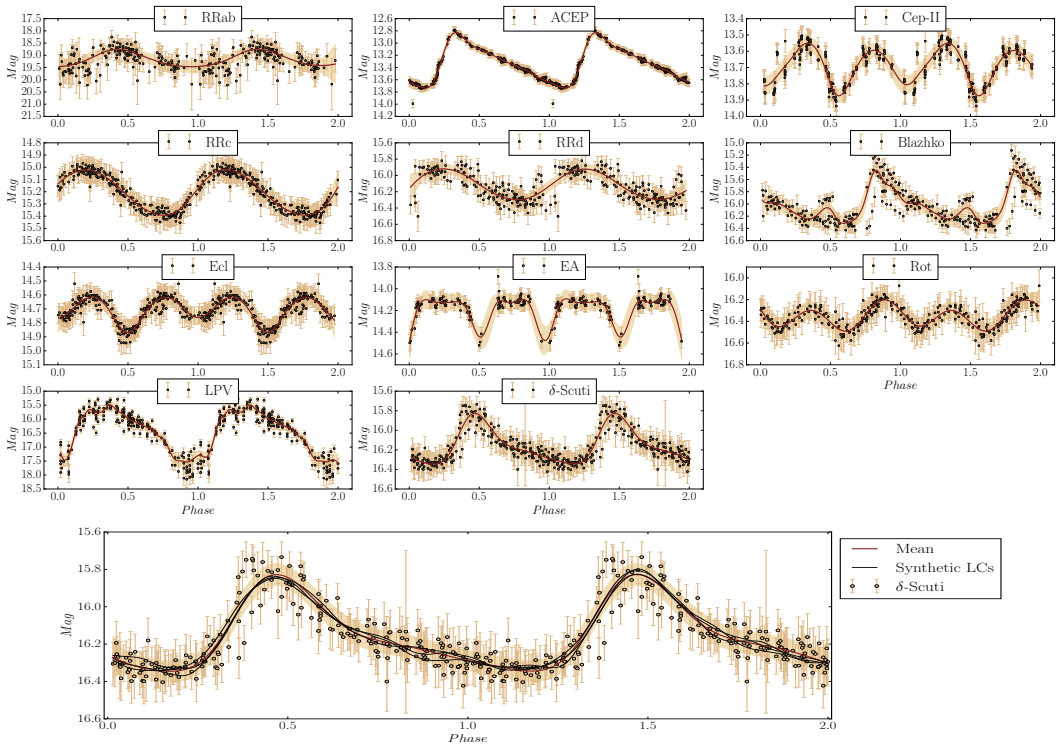

Figure 2: Gaussian Processes offer a flexible approach to produce a smooth model of periodic light curves reported in magnitudes as a function of phase. This is demonstrated with model fits for each example of variable stars considered in the CRTS dataset. The data points are illustrated in black-rounded dots along with the error bars. The mean of the GP fit is shown in brown with three standard deviation away from the mean, shown in shaded pale brown. In the bottom panel, the black lines represent three randomly drawn samples from `GpFit`. These randomly sampled light curves, also known as synthetic LCs together with real LCs, are used in the training set.

synthetic LCs, whilst we test our ML pipeline with only real LCs. These two techniques can be used to oversample both the majority and minority class. The number of training examples after augmentation, *A.Tr* used for each level is given in Fig. 1. Afterwards, features are extracted from these LCs as discussed below.

Stage 3: feature extraction - Our features are based on 6 intrinsic statistical properties relating to location (mean magnitude), scale (standard deviation), variability (mean variance), morphology (skew, kurtosis, amplitude), and time (period). For the `GpFit` and `RASLE` approach, the first six features are extracted directly from the augmented training set (containing both real and synthetic LCs). Whilst for the period feature, the real LCs in the training set are assigned their respective period obtained from CRTS data. A period value is given to each synthetic LC (generated either with `GpFit` or `RASLE`), by randomly sampling from a normal distribution with mean, T (the true period of the real LC from which the synthetic LCs are generated) and within 1 σ-confidence interval, being σ_{T}.

Stage 4: training with Bayesian optimization - We first randomly split our data into 70% training and 30% testing sets. The training set moves through the first level in the HC scheme. The training examples are then augmented using one of the three data augmentation techniques and features are extracted where appropriate. Afterwards, the model (see dotted square at level 1 in Fig. 1) is trained using either RF or XGBoost classifier. The training data are split into fivefolds, where four different folds are kept for training each time and an independent fold is used for validation. We then use a Bayesian optimization approach to find the best hyperparameters for the ML algorithm. Afterwards, we evaluate our trained model based on balanced-accuracy, G-mean, precision, recall, and F1-scores, on real LCs in the test set. The same concepts apply at different levels in the HC where real LCs move down the node, get augmented and classified in their respective classes as shown in Fig. 1.

4 ANALYSIS AND RESULTS

The HC algorithm is trained on both real and artificially augmented data and tested on real data. We show the results of the three data augmentation techniques in Table 2. When using `GpFit` method, we found that our RF implementation performs best at all HC levels (highlighted in Table 2 as grey) when compared to [5], hereafter H19. In addition, we found that both XGBoost and RF provide good performance for variable star classification. In this paper, we assess the consistency of the results using `GpFit` and RF by plotting the Receiver Operator Characteristic (ROC) curve for each class (see Fig. 4). We note that classification performance is very good. The area under the ROC curve (AUC) values are greater than 0.95 for several classes, except for Rotational, RRd, and Blazhko.

We improve upon the result obtained in H19. For instance, the balanced-accuracy increases from 61 to 65 per cent in level 1, from 86 to 88 percent at level 2 for the eclipsing node, from 86 to 87 per cent for subclasses of RR Lyrae at level 3, and finally from 81 to 83 per cent for Cepheids at level 3. To check the consistency and robustness of our new approach, we perform an additional step. We use different splits (K = 5, 6, ...,10) during cross-validation and predict on the 30 percent test set. With these analyses, we obtain an uncertainty on the metric scores considered, for example for Cepheids at level 3, a 0.83 ± 0.02 balanced-accuracy and 0.91 ± 0.01 G-mean score are obtained.

5 Conclusion

In this paper, we present a new approach for tackling the problem of imbalanced data: a level-wise data augmentation in a hierarchical classification framework. Through an empirical investigation, we demonstrate three techniques for augmenting data; that is, `SMOTE`, `RASLE`, and `GpFit` are applied to variable star data. We show that using RF and `GpFit` together can effectively improve recall rates, hence increasing the balanced-accuracy and G-mean scores of the classification pipeline. Our code is made publicly available at https://github.com/Zafiirah13/ICVaS.

Broader Impact

With the advent of large surveys, for instance the MeerLICHT and MeerKAT telescopes, we are entering a time wherein real-time transient surveys are becoming more common. Therefore, it is imperative to prioritize follow-up using light curve classification. Even with the discovery of thousands of transients per night, we are entering a time wherein real-time transient surveys are becoming more common. Our level-wise data augmentation models in a hierarchical regime, designed to overcome this problem, achieves unprecedented accuracy on this challenging task. Our results show that we are capable of discriminating rare examples of subclasses, for example, RR Lyrae and Cepheids light curves. Besides the classification performance, another key concern is the speed. With the proposed pipeline developed in this paper, our algorithm is capable of providing classification probabilities for thousands of light-curves within seconds. Developing such ML based pipeline in Astronomy is helpful in not only reducing the human-labour which are admittedly error-prone but also decreases the computational complexity.

Acknowledgment

ZH acknowledges support from the UK Newton Fund as part of the Development in Africa with Radio Astronomy (DARA) Big Data project delivered via the Science & Technology Facilities Council (STFC). BWS acknowledges funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no. 694745). AM is supported by the Imperial President's PhD Scholarship. VMB acknowledges funding from the National Research Foundation (grant nos 98969 and 119446).

References

Breiman, L. 2001, Machine Learning, 45, 45

Chen, T. & Guestrin, C. 2016, ArXiv e-prints:1603.02754

Drake, A. J., Djorgovski, S. G., Catelan, M., et al. 2017, MNRAS, 469, 3688

Fletcher, R. 1987, Practical methods of optimization

Hosenie, Z., Lyon, R. J., Stappers, B. W., & Mootoovaloo, A. 2019, MNRAS, 488, 4858

Rasmussen, C. E. & Williams, C. K. I. 2006, Gaussian Processes for Machine Learning

Appendix

Table 1: Sample size of classes in CRTS data. The class distribution is extremely imbalanced, such as Ecl are over-represented. we reduced the size of the largest class (Ecl) via random undersampling. We downsample this class from 18803 to 4509 (this makes the number of Ecl examples comparable to the next biggest class, EA) and the remaining Ecl light curves (LCs) are then used for evaluation.

Types of variable stars	**$N_{Objects}$**
RRab (fundamental mode)	4325
RRc (first overtone mode)	3752
RRd (multimode)	502
Blazhko (long-term modulation)	171
Contact & Semi-Detached Binary: Ecl	18803
Detached Binary: EA	4509
Rotational: Rot	3636
Long Period Variable: LPV	1286
δ-Scuti	147
Anomalous Cepheids: ACEP	153
Type-II Cepheids: Cep-II	153

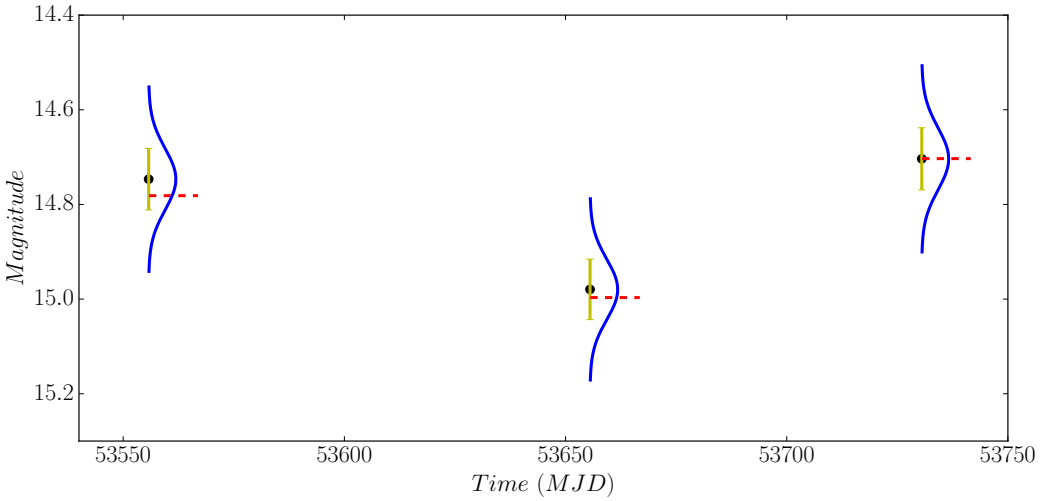

Figure 3: `RASLE`: generating new light curves by random sampling from a normal distribution. The true magnitude along with its error bars is shown in black and yellow. We assume a normal distribution with mean equal to the true magnitude and with sigma equal to the error in magnitude. We randomly draw one sample (red-dashed line) from each normal distribution to produce a completely new light curve.

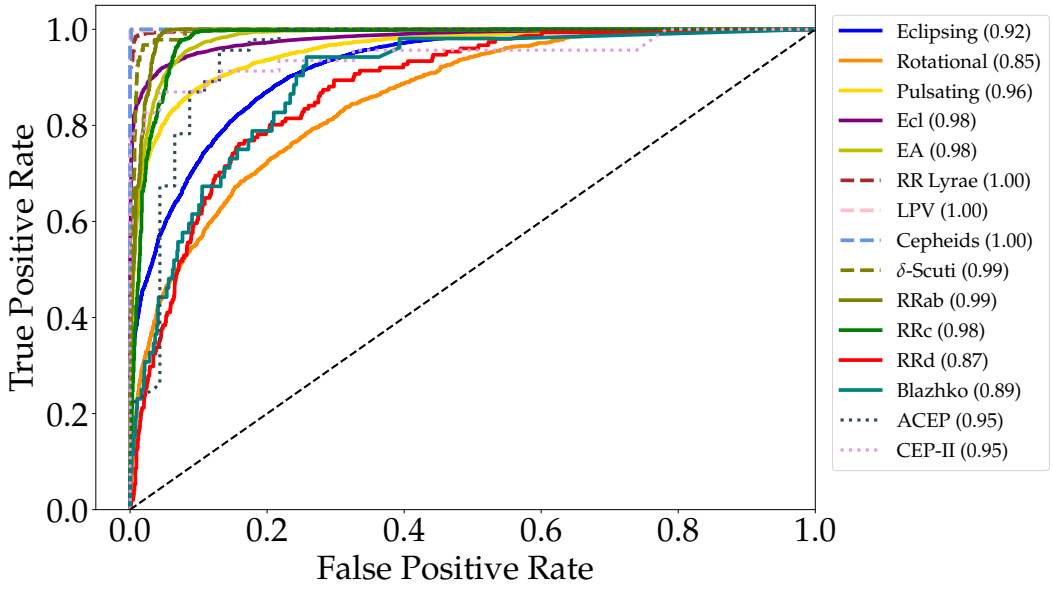

Figure 4: Receiver operating characteristic (ROC) curves for each node in the hierarchical model. Each curve represents a different variable star class with the area under the ROC curve (AUC) score in brackets. This metric is computed on the 30% of the dataset used for testing.

Table 2: Evaluation metrics used to summarize the HC pipeline with the application of three methods of data augmentation. We present the balanced-accuracy and G-mean scores level-wise to evaluate our model. H19 results are presented in bold text in the table. It is seen that the HC pipeline performs fairly well with data augmentation, achieving G-mean scores above $\sim 80\%$ at all levels. The shaded blue represents the augmentation methods that outperform H19. We observe that at all levels, `GpFit` together with RF, performs better than H19 and it is represented in shaded gray. The '$\sim$' represents a single value for the computed average metrics by taking into consideration the overall classes.

Augmentation methods	**Classifiers**	**G-Mean**	**Balanced-accuracy**
First Level: Eclipsing, Rotational and Pulsating Classification			
H19 (No augmentation)	RF	0.78/0.78/0.86 ($\sim$ 0.79)	0.59/0.60/0.75 ($\sim$ 0.61)
SMOTE	XGBoost	0.80/0.77/0.89 ($\sim$ 0.81)	0.63/0.59/0.80 ($\sim$ 0.65)
	RF	0.80/0.78/0.89 ($\sim$ 0.81)	0.63/0.60/0.79 ($\sim$ 0.65)
RASLE	XGBoost	0.82/0.76/0.89 ($\sim$ 0.83)	0.66/0.57/0.79 ($\sim$ 0.68)
	RF	0.82/0.77/0.89 ($\sim$ 0.83)	0.66/0.58/0.79 ($\sim$ 0.68)
GpFit	XGBoost	0.80/0.75/0.89 ($\sim$ 0.81)	0.63/0.56/0.79 ($\sim$ 0.65)
	RF	**0.80/0.75/0.89 ($\sim$ 0.81)**	**0.63/0.56/0.78 ($\sim$ 0.65)**
Second Level: RR Lyrae, LPV, Cepheids and δ-Scuti			
H19 (No augmentation)	RF	0.99/1.00/0.97/1.00 ($\sim$ 0.99)	0.98/0.99/0.93/1.00 ($\sim$ 0.98)
SMOTE	XGBoost	0.99/1.00/1.00/0.95 ($\sim$ 0.99)	0.97/0.99/1.00/0.90 ($\sim$ 0.97)
	RF	0.99/1.00/1.00/0.96 ($\sim$ 0.99)	0.97/0.99/1.00/0.92 ($\sim$ 0.97)
RASLE	XGBoost	0.99/1.00/0.99/0.93 ($\sim$ 0.99)	0.98/1.00/0.98/0.85 ($\sim$ 0.98)
	RF	0.99/1.00/1.00/0.94 ($\sim$ 0.99)	0.98/0.98/1.00/0.88 ($\sim$ 0.98)
GpFit	XGBoost	0.99/1.00/0.99/0.95 ($\sim$ 0.99)	0.97/0.99/0.97/0.99 ($\sim$ 0.98)
	RF	**0.99/1.00/1.00/0.97 ($\sim$ 0.99)**	**0.97/0.99/1.00/0.93 ($\sim$ 0.98)**
Second Level: Ecl and EA			
H19 (No augmentation)	RF	0.93/0.93 ($\sim$ 0.93)	0.86/0.86 ($\sim$ 0.86)
SMOTE	XGBoost	0.94/0.94 ($\sim$ 0.94)	0.88/0.88 ($\sim$ 0.88)
	RF	0.94/0.94 ($\sim$ 0.94)	0.88/0.88 ($\sim$ 0.88)
RASLE	XGBoost	0.93/0.93 ($\sim$ 0.93)	0.85/0.85 ($\sim$ 0.85)
	RF	0.93/0.93 ($\sim$ 0.93)	0.85/0.86 ($\sim$ 0.86)
GpFit	XGBoost	0.93/0.93 ($\sim$ 0.93)	0.88/0.88 ($\sim$ 0.88)
	RF	**0.94/0.94 ($\sim$ 0.94)**	**0.87/0.88 ($\sim$ 0.88)**
Third Level: RRab, RRc, RRd and Blazhko			
H19 (No augmentation)	RF	0.97/0.92/0.65/0.44 ($\sim$ 0.92)	0.94/0.85/0.40/0.18 ($\sim$ 0.86)
SMOTE	XGBoost	0.95/0.92/0.67/0.58 ($\sim$ 0.91)	0.91/0.83/0.42/0.31 ($\sim$ 0.83)
	RF	0.95/0.82/0.47/0.33 ($\sim$ 0.91)	0.91/0.82/0.47/0.33 ($\sim$ 0.83)
RASLE	XGBoost	0.96/0.95/0.56/0.53 ($\sim$ 0.92)	0.93/0.89/0.30/0.26 ($\sim$ 0.87)
	RF	0.97/0.95/0.52/0.52 ($\sim$ 0.92)	0.94/0.90/0.25/0.25 ($\sim$ 0.87)
GpFit	XGBoost	0.97/0.93/0.57/0.44 ($\sim$ 0.92)	0.94/0.86/0.30/0.17 ($\sim$ 0.85)
	RF	**0.97/0.93/0.56/0.41 ($\sim$ 0.92)**	**0.94/0.87/0.32/0.26 ($\sim$ 0.87)**
Third Level: ACEP and Cep-II			
H19 (No augmentation)	RF	0.90/0.90 ($\sim$ 0.90)	0.82/0.80 ($\sim$ 0.81)
SMOTE	XGBoost	0.88/0.88 ($\sim$ 0.88)	0.78/0.76 ($\sim$ 0.77)
	RF	0.88/0.88 ($\sim$ 0.88)	0.78/0.76 ($\sim$ 0.77)
RASLE	XGBoost	0.88/0.88 ($\sim$ 0.88)	0.77/0.78 ($\sim$ 0.77)
	RF	0.88/0.88 ($\sim$ 0.88)	0.77/0.78 ($\sim$ 0.78)
GpFit	XGBoost	0.88/0.88 ($\sim$ 0.88)	0.78/0.78 ($\sim$ 0.78)
	RF	**0.91/0.91 ($\sim$ 0.91)**	**0.84/0.82 ($\sim$ 0.83)**