Machine Learning Methods Applied to Star Formation Classification: A Report

B. Crompvoets

May 23, 2022

1 Introduction

This work will address the issue of classifying young stellar objects (YSOs) given input from a field of view from the Spitzer telescope. YSOs are defined as any stellar object which has not yet reached Main Sequence evolution. In this regime, there are four stages known as the Class 0, Class I, Class II, and Class III stages. Class 0 is the label given to the youngest YSOs. In particular, these objects have only just begun to collapse from their natal dust cloud, and are still heavily obscured. They are actively accreting dust and gas and thus gaining mass. Their luminosity peaks deep in the infrared (IR), a wavelength regime not well characterized by Spitzer. Class I YSOs are those slightly older than Class 0. Their dust and gas envelope has begun to thin, but they are still obscured. Due to this, they also are located in the IR, but not to the same extent as the Class 0s, and are thus within the detectability range of Spitzer. Class II YSOs no longer have an envelope, but instead are surrounded by a disk of material. These are the stars often imaged when searching for planet formation in these debris disks. Finally, Class III YSOs are the eldest. Their disk has dissipated, and they are condensing as they near the point of Hydrogen ignition in their cores. Not to be unaddressed, a final classification may be made to include flat-spectrum YSOs, so called because the slope of their spectral energy distribution is approximately zero.

The objects of greatest interest in this study are the youngest of these YSOs, as probing their formation can give us information on the kinematics and dynamics of star formation. In order to get a full sense of the process, we must be able to rely upon several samples of YSOs across all stages. Thus, the initial objective must be to obtain a reliable categorization of objects identified in surveys. This objective was fulfilled by Gutermuth et al. (2008) and updated by Gutermuth et al. (2009, hereafter G09). They provided a prescription for categorizing objects given the four Spitzer IRAC bands (located at 3.6μm, 4.5μm, 5.8μm, and 8.0μm),the 24μm Spitzer MIPS band, and the J, H, and K_S 2MASS bands. They apply empirical cuts onto a range of colour-colour and colour-magnitude diagrams in order to separate stellar objects into a number of classes. Although a very popular method to use, this method is time-expensive and requires heavy supervision. In the era of big data science, such methods are becoming archaic, and thus new methods must be developed to handle the large input of data. The modern solution to this problem is the use of machine learning, or ML.

We refer to two groups who have attempted to use ML methods to classify YSOs: Miettinen (2018) and Cornu and Montillaud (2021, hereafter CM21). Miettinen (2018) use a compilation of various ML methods in order to classify YSOs into three categories: Class 0, Class I, and flat spectrum. They test eight different methods, seven classical methods, and one multilayer perceptron (MLP), many of the same methods we test again here. They found that the best method to separate the classes from one another was a gradient boosting model. As they used supervised models, they had to have a classification pre-applied to their data, and this was done by using the data-set from the Orion Protostar Survey, as classified by Furlan et al. (2016). Although providing fair results, the major shortcoming of this work is that they only consider proto-stars, and thus they can only classify objects into their YSO category if they already know the object is a YSO. A more realistic approach is to create a method which can classify YSOs based off of the unsorted input from a telescope. This is performed by CM21, who use an MLP to classify objects from Spitzer as either Class I YSOs, Class II YSOs, or contaminant objects. The reason for using only the three classes is that (1) Class 0 objects are not visible in Spitzer data (the only data used in this survey); and (2) Class III objects have a signature very similar to regular main sequence stars at these wavelengths, and thus the neural network would lose performance in trying to correctly classify Class III. As the most important YSOs to star formation re-

search is the youngest of the categories, it is then a reasonable sacrifice to allow any Class III or flat spectrum YSOs to be classified as contaminants. They test their network on the full Orion Survey (Megeath et al., 2012), as well as on a survey of NGC 2264 (Rapson et al., 2014). These surveys include a number of different object types, and the training and validation sets are specifically chosen to increase model performance while still being representative of the night sky (see Section 2).

We then seek to combine the best of each of these works in our own formulation. In particular, we use the same data and targets as CM21, as well as three different data-splits. CM21 claims that the predictions from their method may be closer to ground truth than the targets. Thus, we follow their suggestion to train on their results and compare the two target methods in order to test this claim. Our aim is to perform the same classification as CM21, but using a number of classical methods as well as a simpler MLP.

The layout of this report is as follows: Section 2 outlines the data, the targets, and the data-split, as well as the different methods we make use of. Section 3 describes the best results across all methods for each of the targets used. Section 4 discusses the results obtained; and finally, Section 5 summarizes the report, our conclusions, and future outlook.

2 Data and Methodology

2.1 Data

As mentioned above, the data obtained is from a collection of Spitzer surveys, namely Orion (Megeath et al., 2012) and NGC 2264 (Rapson et al., 2014). We use two sets of targets when training: "G-targets", and "C-targets". G-targets are the object classifications as determined via a simplified version of the method from G09. Conversely, C-targets are the predictions from CM21. If CM21 are correct in their claim that their predictions are closer to ground truth than those from the simplified version of G09 we would expect to see higher performance across all metrics with the C-targets as compared to the G-targets. We also will only be using data from the four Spitzer IRAC bands, as well as their errors, as the MIPS band is sparsely populated in the sample, a problem that classical ML methods cannot surpass. For ease of comparison and to ensure the methods are focussed on classifying our youngest YSOs, we follow CM21 in having only three classes: Class I YSOs, Class II YSOs, and Contaminant objects.

In machine learning, the preferred method of training is to use a training set, a validation set, and a test set. The validation set is used to monitor performance as the network trains, whereas the test set is completely withheld from any training and is used to see if the network behaves well when presented with new data. As we are dealing with extremely imbalanced data in our three classes with the class of greatest interest having only $\sim$400, we must neglect the use of a test set and instead follow the common practice of using our validation set as our test set.

Furthermore, we use three different data-splits in order to test which method works best. The exact splits of these sets are presented in Table 1. The CM21-split comes from Cornu and Montillaud (2021), which we will explain briefly here. First, the test set numbers are chosen to reflect the ratios that were initially returned in the survey of each of the seven sub-classes, and the value of that ratio is determined via the hyper-parameter θ. Secondly, the training set is optimized with a second hyper-parameter for each sub-class. They begin by assigning however many Class I YSOs are left after removing the test split as the number of Class I YSOs in the training set, as this is the most diluted class of interest. Next, they assign a fraction to each sub-class, which is determined as the number of objects in a given class (N_i^{TR}) over the number of objects in Class I (N_{CI}^{TR}), or $\gamma_i = \frac{N_i^{TR}}{N_{CI}^{TR}}$. These two hyper-parameters, θ and γ, are optimized to achieve the best results for their MLP, and we utilize these values for accurate comparison. The 300s-split uses 300 of every sub-class which has 300 objects, the remaining amount from CM21 for those that did not. The test set for this split is the same as CM21. Finally, the 75/25 split is a basic split in which 75% of each class is used in the training and 25% is used in the test set.

2.2 Methodology

With the data in-hand, we now determine the best algorithms for the classifications. As each method has a number of parameters, we use an optimization method known as GridSearch. GridSearch tests over a user-defined parameter space with the given model, until a set of optimized paramaters is determined. It uses a complicated function akin to Mean Squared Error, although the specific metric can be chosen. We leave the metric as is, and perform the GridSearch for five of the following six methods.

Our first classical method tested is that of a Support Vector Machine or SVM. These machines take the input data and separate it into higher dimen-

	CM21		300s		75/25	
Type	Train	Test	Train	Test	Train	Test
Class I	331	82	300	82	311	103
Class II	1141	531	300	531	1994	665
Galaxies	231	104	300	104	391	130
AGNs	529	278	300	278	1043	348
Shocks	27	6	27	6	25	9
PAHs	70	17	70	17	66	22
Stars	1257	4359	300	4359	21796	5449

Table 1: The split of each class into the numbers in training and test sets for each data-split tested.

sional space such that the classes can be separated. Similarly, a Logisitic Regressor (LR) attempts to define a function to linearly separate classes in the n-dimensional space of the feature input. These methods, in theory, are most similar to the empirical cuts performed by G09, but instead they perform their cuts in a higher dimensional space and use fluxes, not magnitudes. We then move on to more complicated methods called ensembles. The first ensemble is a stacking ensemble (Stack) wherein the previous two methods are combined and the machine finds the best way for them to work together. The second ensemble method we use is known as Random Forest, or RF. Our RF makes use of 100 decision trees, using the mode of all trees to determine the classification of each object. Gradient Boosting (GB) machines also use decision trees, but in this case each tree is performed consecutively, so that the next tree can learn from the mistakes of the previous. A step above this is eXtreme Gradient Boosting (XGBoost or XGB), where in addition to GB, the network also performs error optimization, forcing the next tree to more aggressively solve where the previous tree went wrong. Each method has benefits and drawbacks, and the best tool for the job is rarely clear. To recall, Miettinen (2018) used SVM, LR, regular decision trees, RF, and GB, as well as an MLP, and they found that a GB machine was the best method, however they only sorted YSOs with no contaminant classes.

Outside of using classical methods, we also used a MLP. We follow CM21 in their formulation, with one input layer, one hidden layer, and an output layer. As we are only using the Spitzer IRAC bands and their associated errors, we have eight input neurons. For the hidden layer, we use the same number of neurons as CM21, who used 20 neurons. They chose this value from the following formulation: first, they began with twice as many neurons as number of cuts applied when following the G09 method; second, they increased this value until the network performance ceased increasing. Lastly, our output layer contains only three neurons, which will hold the probability of each class, where the class with the greatest probability is assigned to the object for the networks prediction. For our network parameters, we follow those given in CM21, differing in three respects: (1) we use a batch size of 25, they specify they use batch training but neglect to mention their batch size; (2) we use a learning rate of $4e{-}2$ in comparison to their learning rate of $4e{-}5$; and (3) we run for only 10 000 epochs as a consequence of our more aggressive learning rate.

3 Results

We use recall as our metric of choice as we wish to maximize the number of objects correctly classified of all three classes, thereby minimizing the number of false negatives. For all metrics recorded (recall, precision, and accuracy) for all runs, see Appendix A. As both of these YSO classes are of great interest to star formation research, we choose those algorithms for which Class I has the highest recall, using the recall from Class II as a tie-breaker. In some cases, generally with the 300s-split, Class II objects tend to have a much lower recall than in other methods. We find that across all algorithms, both the CM21-split and the 300s-split perform extremely well. When using the CM21-split, Class II objects have an increased performance in comparison to with using the 300s split, but Class I objects often have a decreased performance. However, this trade-off was deemed preferential, as the decrease in recall in Class I was often less than the increase in Class II, and as Class II is the class with a larger number of objects, this lead to greater precision and accuracy overall We present our best results in Figures 1 and 2, where all algorithms use the CM21-split, although the 300s-split is also a reasonable method.

We find that using the predictions from CM21 does lead to higher performance across all algorithms. Indeed, we see an increase of as much as 10% in recall in each class in some cases. However, in other cases,

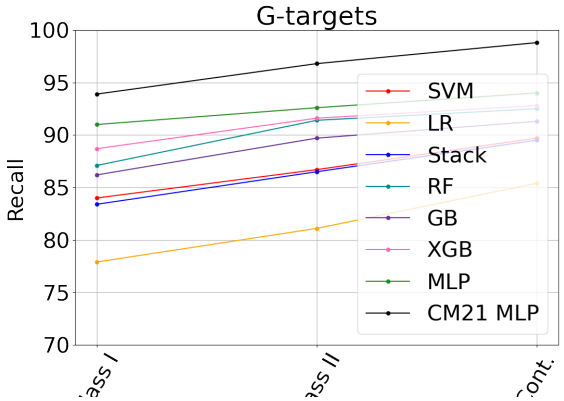

Figure 1: The recall for each class for each "best" algorithms as tested with the G-targets. SVM, LR, and Stack make use of the 300s data-split, and RF, GB, XGB, and the MLP all make use of the CM21-split. 's results are included for comparison.

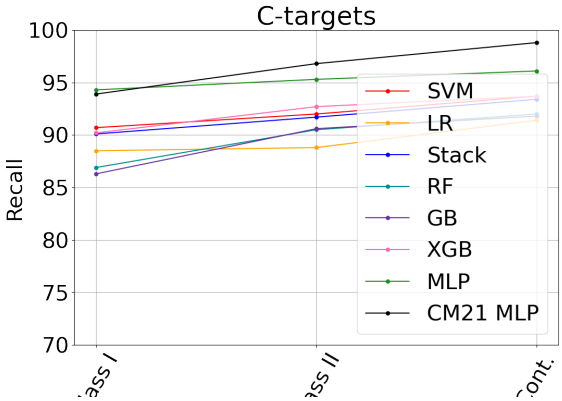

Figure 2: The recall for each class for each "best" algorithms as tested with the C-targets. SVM, LR, and Stack make use of the 300s data-split, and RF, GB, XGB, and the MLP all make use of the CM21-split.'s results are included for comparison, they were trained on G-targets.

the methods perform equally well with G-targets as C-targets, and in all cases the performance is within errors. Each method was run 200 times with a differing random seed in order to collect mean and standard deviations for the methods in order to obtain comparable estimates. Errors are available in Appendix A.

We choose our best models as the ones that perform best using G-targets. These targets are readily available and widely used, making them far more useful when we move to classifying data from other surveys. As well, CM21 only provides three classes for their predicting labels, leading to a greater uncertainty in which sub-classes are being sampled on, and does not allow for us to determine which sub-classes are being incorrectly classified. Thus, using G-targets, our best method across all metrics and classes was the MLP. Although simple with an aggressive learning rate, it outperforms all other models, and has a smaller standard deviation in its results to XGBoost, which comes in second. These top two methods have performance within less than a standard deviation of one another, allowing us to compare them in greater depth.

4 Discussion

As both top models perform (nearly) equally well, we will probe further into where exactly the XGBoost misclassified objects, and which objects (including subclasses) were misclassified. G09 specify that in their work, contamination of the Class I and Class II YSO classes may arise from non-YSO sources with excess infrared emission. These sources are star-forming galaxies, broad-line active galactic nuclei (AGNs), and unresolved knots of shock emission from outflows colliding with cold cloud material (Gutermuth et al., 2009). However, Figure 3 shows that the vast majority of misclassifications are with Class II YSOs and regular stars, and most of the objects incorrectly classified as Class I are actually Class II objects. This second misclassification is likely due to the missing MIPS 24μm band. This band is of particular importance, as it separates out the reddened Class II YSOs from Class I. This band is available for only 15% of data, with most but not all Class I and Class II objects having data. The incorporation of this band will be considered in future work. In addition, the contamination in Class II is due almost purely to regular Main Sequence stars or possibly Class III YSOs (which have been included in the Stars sub-class). This latter may also explain the number of Class II objects incorrectly classified as contaminants. As YSOs evolve from Class II to Class III, their signatures become more similar, making it possible that older Class II YSOs are being mis-classified as Contaminants, and younger Class III or Flat-spectrum YSOs are being mis-classified as Class II. The original surveys that this data was pulled from involved the full G09 treatment of objects, which outlined which objects were Class III/Flat-spectrum. The identification of these YSOs will be addressed in future work. It is important to note that only $\sim 4.5\%$ of the total star subclass was mis-classified as either Class I or Class II, and with this sub-class being extremely varied, it is most likely to be prone to over-lapping the

input space in which our objects of interest reside.

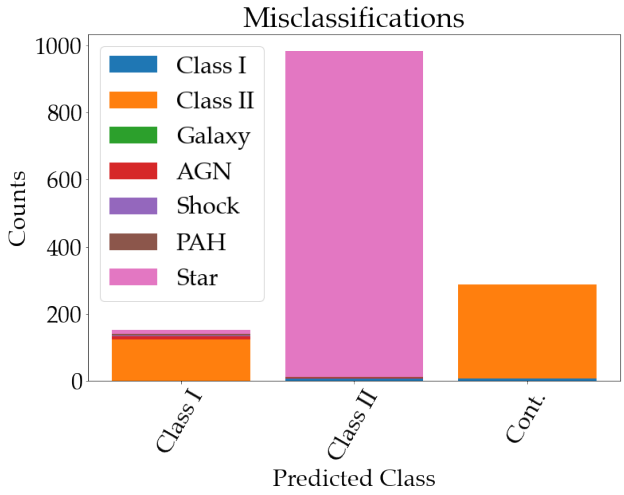

Figure 3: The predicted class, as determined using an XGBoost trained with the CM21-split, forwarded on the full data-split, shows which objects tend to get misclassified. For instance, many Class II objects are incorrectly classified as either Class I or Contaminants in this method.

We find that in using C-targets instead of G-targets that we achieve higher metrics across all classes. This seems to support the claim by CM21 that their results are closer to ground truth than the simplified G09 method. However, it is unclear what the results may have been had the targets had the full G09 method been implemented instead. Further analysis is needed. The reason the full G09 method was not used was because it relied upon information from other telescopes than only Spitzer, and thus relied on unavailable data to classify. Furthermore, the simplified method allowed the CM21 team to create their own list of targets without having to work around the different empirical cuts different teams may have used. As the goal of using ML methods is to be able to work around missing data and to work with new data, this may have been a too-cautious approach.

5 Conclusions

We performed a classification analysis of Spitzer data from the Orion molecular cloud, as well as NGC 2264 using six different classical ML methods and one MLP. We find that XGBoost and the MLP perform nearly equivalently, with the MLP slightly outperforming XGBoost. The majority of misclassifications occur due to missing data which would aid in the separation of Class I from Class II YSOs, and from the lack of a Class III/Flat-spectrum YSO class. These misclassifications may be rectified in future work. Overall, the top two methods performed very closely to the MLP from CM21, from whom our MLP is based upon and our data drawn from.

In future work, we wish to apply these methods to classifying all sub-classes possible using the full G09 method, since, in addition to highlighting Class III/Flat-spectrum YSOs, these classifications will be helpful to researchers in other fields as well. We will also attempt to classify using the MIPS 24μm band. A further work beyond this is to explore a slightly more complex MLP, as well as expanding these methods to multiple surveys. As each new survey will have different data input, there is the possibility that we can make use of what is known as transfer learning, in which the initially trained MLP is updated with the new data. Alternatively, we may retrain on the data from every new telescope, until we have obtained enough methods that we can create a program which, given the telescope and bands in use, will assign classes to all objects in an input dataset.

A Precision, Recall and Accuracy Tables for each run

In the following tables, we list the results of every run, each table for each algorithm tested.

References

D. Cornu and J. Montillaud. A neural network-based methodology to select young stellar object candidates from IR surveys. , 647:A116, Mar. 2021. doi: 10.1051/0004-6361/202038516.

E. Furlan, W. J. Fischer, B. Ali, A. M. Stutz, T. Stanke, J. J. Tobin, S. T. Megeath, M. Osorio, L. Hartmann, N. Calvet, C. A. Poteet, J. Booker, P. Manoj, D. M. Watson, and L. Allen. The Herschel Orion Protostar Survey: Spectral Energy Distributions and Fits Using a Grid of Protostellar Models. , 224(1):5, May 2016. doi: 10.3847/0067-0049/224/1/5.

R. A. Gutermuth, P. C. Myers, S. T. Megeath, L. E. Allen, J. L. Pipher, J. Muzerolle, A. Porras, E. Winston, and G. Fazio. Spitzer Observations of NGC 1333: A Study of Structure and Evolution in a Nearby Embedded Cluster. , 674(1):336–356, Feb. 2008. doi: 10.1086/524722.

		G-targets			C-targets		
Split	Class	Recall	Precision	Accuracy	Recall	Precision	Accuracy
CM21	Class I	84.0 ± 9.4	79.6 ± 13.6		90.7 ± 4.8	78.5 ± 16.5	
CM21	Class II	86.7 ± 8.0	84.6 ± 11.4	96.8 ± 0.0	92.0 ± 4.2	85.5 ± 13.9	97.4 ± 0.0
CM21	Cont.	89.7 ± 9.5	87.6 ± 13.0		93.7 ± 4.9	87.7 ± 15.1	
300s	Class I	87.4 ± 9.9	68.7 ± 16.4		93.0 ± 9.0	67.0 ± 27.6	
300s	Class II	81.8 ± 11.6	68.7 ± 16.4	91.7 ± 0.0	87.7 ± 10.3	79.9 ± 23.5	95.2 ± 0.0
300s	Cont.	87.8 ± 10.1	78.2 ± 18.9		92.6 ± 8.8	81.6 ± 24.5	
75/25	Class I	76.8 ± 15.1	83.6 ± 6.8		82.9 ± 11.2	88.3 ± 9.1	
75/25	Class II	83.0 ± 12.6	87.7 ± 7.1	88.6 ± 0.0	86.2 ± 9.6	89.8 ± 9.9	88.3 ± 0.0
75/25	Cont.	85.4 ± 14.0	85.0 ± 6.0		89.7 ± 11.3	83.7 ± 10.8	

Table 2: Recall, precision, and accuracy for each of the different data splits and target sources on using SVM.

		G-targets			C-targets		
Split	Class	Recall	Precision	Accuracy	Recall	Precision	Accuracy
CM21	Class I	77.9 ± 12.4	79.1 ± 10.9		88.5 ± 4.8	74.0 ± 16.2	
CM21	Class II	81.1 ± 10.7	80.2 ± 10.3	95.1 ± 0.0	88.8 ± 4.6	78.9 ± 13.9	95.5 ± 0.0
CM21	Cont.	85.4 ± 12.77	85.7 ± 12.2		91.4 ± 5.4	83.8 ± 16.3	
300s	Class I	81.6 ± 9.1	70.2 ± 19.2		91.8 ± 11.2	67.6 ± 25.5	
300s	Class II	76.8 ± 10.8	62.9 ± 22.7	87.2 ± 0.0	85.2 ± 12.9	78.9 ± 21.6	94.8 ± 0.0
300s	Cont.	82.7 ± 9.7	76.4 ± 22.3		91.3 ± 11.1	81.3 ± 22.9	
75/25	Class I	68.5 ± 11.2	73.2 ± 6.6		75.7 ± 13.4	82.2 ± 15.4	
75/25	Class II	70.8 ± 9.9	76.8 ± 7.8	75.1 ± 0.0	75.7 ± 13.4	83.1 ± 16.0	75.9 ± 0.0
75/25	Cont.	75.3 ± 11.8	72.6 ± 6.9		83.4 ± 15.5	73.6 ± 18.1	

Table 3: Recall, precision, and accuracy for each of the different data splits and target sources on using LR.

R. A. Gutermuth, S. T. Megeath, P. C. Myers, L. E. Allen, J. L. Pipher, and G. G. Fazio. A Spitzer Survey of Young Stellar Clusters Within One Kiloparsec of the Sun: Cluster Core Extraction and Basic Structural Analysis. , 184(1):18–83, Sept. 2009. doi: 10.1088/0067-0049/184/1/18.

S. T. Megeath, R. Gutermuth, J. Muzerolle, E. Kryukova, K. Flaherty, J. L. Hora, L. E. Allen, L. Hartmann, P. C. Myers, J. L. Pipher, J. Stauffer, E. T. Young, and G. G. Fazio. The Spitzer Space Telescope Survey of the Orion A and B Molecular Clouds. I. A Census of Dusty Young Stellar Objects and a Study of Their Mid-infrared Variability. , 144(6):192, Dec. 2012. doi: 10.1088/0004-6256/144/6/192.

O. Miettinen. Protostellar classification using supervised machine learning algorithms. , 363(9):197, Sept. 2018. doi: 10.1007/s10509-018-3418-7.

V. A. Rapson, J. L. Pipher, R. A. Gutermuth, S. T. Megeath, T. S. Allen, P. C. Myers, and L. E. Allen. A Spitzer View of the Giant Molecular Cloud MON OB1 EAST/NGC 2264. , 794(2):124, Oct. 2014. doi: 10.1088/0004-637X/794/2/124.

		G-targets			C-targets		
Split	Class	Recall	Precision	Accuracy	Recall	Precision	Accuracy
CM21	Class I	83.4 ± 10.0	83.5 ± 10.0		90.1 ± 5.3	82.5 ± 11.7	
CM21	Class II	86.5 ± 8.5	86.5 ± 8.5	97.0 ± 0	91.7 ± 4.6	86.6 ± 9.9	97.4 ± 0.0
CM21	Cont.	89.5 ± 10.0	89.5 ± 10.0		93.4 ± 5.4	89.4 ± 11.3	
300s	Class I	86.8 ± 9.6	70.3 ± 15.5		93.1 ± 9.1	69.1 ± 25.4	
300s	Class II	81.5 ± 11.3	69.4 ± 16.0	91.8 ± 0.0	87.8 ± 10.4	80.8 ± 21.6	95.5 ± 0
300s	Cont.	87.5 ± 10.0	78.9 ± 18.1		92.8 ± 8.9	82.6 ± 22.6	
75/25	Class I	76.0 ± 16.1	84.3 ± 6.2		82.4 ± 11.7	88.2 ± 9.2	
75/25	Class II	82.6 ± 13.6	88.0 ± 6.7	88.6 ± 0.0	86.0 ± 10.0	89.7 ± 10.0	88.2 ± 0.0
75/25	Cont.	85.1 ± 14.9	85.3 ± 5.6		89.5 ± 11.7	83.6 ± 10.9	

Table 4: Recall, precision, and accuracy for each of the different data splits and target sources on using Stacking.

		G-targets			C-targets		
Split	Class	Recall	Precision	Accuracy	Recall	Precision	Accuracy
CM21	Class I	87.1 ± 8.6	84.6 ± 8.8		86.9 ± 9.0	86.9 ± 7.3	
CM21	Class II	91.4 ± 6.0	86.3 ± 7.6	97.2 ± 0.1	90.5 ± 8.1	87.5 ± 6.9	97.4 ± 0.1
CM21	Cont.	92.5 ± 7.2	90.4 ± 9.0		92.0 ± 8.2	91.3 ± 8.0	
300s	Class I	87.0 ± 4.7	70.9 ± 16.3		91.6 ± 7.1	75.4 ± 19.1	
300s	Class II	87.4 ± 5.0	70.9 ± 16.4	93.6 ± 0.3	88.5 ± 8.1	83.9 ± 16.5	96.8 ± 0.1
300s	Cont.	90.7 ± 4.6	80.7 ± 18.3		92.6 ± 8.4	85.7 ± 17.1	
75/25	Class I	79.7 ± 13.7	86.2 ± 6.2		82.6 ± 13.0	89.0 ± 5.1	
75/25	Class II	86.0 ± 11.8	88.7 ± 7.5	91.1 ± 0.2	87.9 ± 10.5	90.3 ± 5.9	90.9 ± 0.2
75/25	Cont.	87.7 ± 12.7	87.4 ± 5.7		90.4 ± 11.4	87.5 ± 6.0	

Table 5: Recall, precision, and accuracy for each of the different data splits and target sources on using RF.

		G-targets			C-targets		
Split	Class	Recall	Precision	Accuracy	Recall	Precision	Accuracy
CM21	Class I	86.2 ± 9.0	85.7 ± 7.8		86.3 ± 9.1	85.6 ± 8.0	
CM21	Class II	89.7 ± 7.8	86.1 ± 7.6	97.1 ± 0.0	90.6 ± 7.5	86.3 ± 7.5	97.1 ± 0.0
CM21	Cont.	91.3 ± 8.6	90.3 ± 8.9		91.8 ± 8.1	90.4 ± 8.9	
300s	Class I	85.3 ± 5.9	68.4 ± 17.7		90.9 ± 8.0	75.0 ± 18.9	
300s	Class II	84.7 ± 6.0	$68.9pm17.5$	92.7 ± 0.8	86.6 ± 9.3	82.4 ± 16.2	96.4 ± 0.2
300s	Cont.	88.3 ± 6.6	79.8 ± 18.6		92.1 ± 8.2	85.7 ± 16.9	
75/25	Class I	79.0 ± 14.2	82.9 ± 8.8		81.6 ± 13.2	86.2 ± 6.0	
75/25	Class II	85.1 ± 12.0	88.0 ± 8.3	90.3 ± 0.0	86.8 ± 11.1	89.1 ± 7.2	89.9 ± 0.0
75/25	Cont.	87.0 ± 13.0	86.2 ± 7.4		89.2 ± 12.4	85.0 ± 6.7	

Table 6: Recall, precision, and accuracy for each of the different data splits and target sources on using GB.

		G-targets			C-targets		
Split	Class	Recall	Precision	Accuracy	Recall	Precision	Accuracy
CM21	Class I	88.7 ± 7.2	84.6 ± 9.1		90.2 ± 6.2	87.2 ± 7.1	
CM21	Class II	91.6 ± 6.0	86.7 ± 8.0	97.4 ± 0.0	92.7 ± 5.2	87.9 ± 6.7	97.5 ± 0.0
CM21	Cont.	92.8 ± 6.7	90.2 ± 9.5		93.7 ± 5.7	91.5 ± 8.0	
300s	Class I	89.4 ± 4.2	73.8 ± 14.6		91.4 ± 6.3	76.6 ± 18.3	
300s	Class II	88.2 ± 4.8	75.3 ± 13.7	94.4 ± 0.0	88.6 ± 7.5	84.6 ± 15.4	97.0 ± 0.0
300s	Cont.	91.1 ± 4.9	82.6 ± 16.2		92.9 ± 7.2	86.6 ± 16.6	
75/25	Class I	84.4 ± 9.9	89.0 ± 4.2		86.1 ± 9.8	90.0 ± 4.6	
75/25	Class II	88.5 ± 8.3	91.6 ± 4.5	92.2 ± 0.0	89.9 ± 8.2	91.7 ± 5.4	92.2 ± 0.0
75/25	Cont.	90.1 ± 9.2	89.9 ± 3.7		91.7 ± 9.2	88.5 ± 5.3	

Table 7: Recall, precision, and accuracy for each of the different data splits and target sources on using XGB.

		G-targets			C-targets		
Split	Class	Recall	Precision	Accuracy	Recall	Precision	Accuracy
CM21	Class I	91.0 ± 4.9	84.6 ± 9.4		94.3 ± 3.0	89.9 ± 5.8	
CM21	Class II	92.6 ± 4.2	87.2 ± 8.2	97.6 ± 0.0	95.3 ± 2.6	91.0 ± 5.2	98.2 ± 0.0
CM21	Cont.	94.0 ± 4.9	90.4 ± 9.7		96.1 ± 3.0	93.5 ± 6.2	
300s	Class I	90.2 ± 4.8	80.8 ± 13.2		95.5 ± 2.4	76.8 ± 17.0	
300s	Class II	91.0 ± 4.4	85.6 ± 11.1	97.3 ± 0.0	94.2 ± 2.9	83.5 ± 14.3	97.1 ± 0.0
300s	Cont.	93.2 ± 5.3	88.5 ± 12.6		95.8 ± 2.7	86.6 ± 15.9	
75/25	Class I	86.0 ± 8.6	86.7 ± 7.2		91.9 ± 5.1	92.2 ± 4.2	
75/25	Class II	89.4 ± 7.2	91.0 ± 7.0	92.6 ± 0.0	93.4 ± 4.3	93.9 ± 5.0	94.0 ± 0.0
75/25	Cont.	90.9 ± 8.0	89.2 ± 6.0		95.0 ± 5.1	90.9 ± 4.9	

Table 8: Recall, precision, and accuracy for each of the different data splits and target sources on using the MLP.