UNIVERSITY OF EXETER

Computer Science Department

MSci. Computer Science & Mathematics
Computer Science Department

Using Machine Learning and Clustering Techniques to Identify Globular Cluster Candidates in the Halo of M31

Candidate

Judy Warner-Willich

Student ID 245612

Supervisor

Dr. Avon Huxor

University of Exeter

Co-supervisor

Academic Year
2022/2023

Abstract

Globular clusters (GCs) are ancient clusters of stars that exist in and around galaxies. Their properties shed light on many questions in astronomy and so identification of them can have a great impact on many areas of study. In this work we implement two machine learning (ML) classification algorithms - namely random forest and multilayer perceptron - and combine them with the clustering algorithm DBSCAN to identify possible GC candidates in the galactic halo of M31, and compare the ML models' abilities to do so. Photometry from the Pan-Andromeda Archaeological Survey (PAndAS) matched with a catalogue of known M31 GCs and galaxies were used for training. Our random forest model scores 83.5% ± 0.04% in accuracy and 83.6% ± 0.06% recall, and multilayer perceptron performed with 82.7% ± 0.04% accuracy and a higher recall of 88.0% ± 0.06%. We also present some GC candidates to demonstrate the real application of the models to data from PAndAS.

	Yes	No
I certify that all material in this dissertation which is not my own work has been identified.	☑	☐
I give the permission to the Department of Computer Science of the University of Exeter to include this manuscript in the institutional repository, exclusively for academic purposes.	☑	☐

Contents

List of Figures

List of Tables

List of Algorithms

List of Code Snippets

List of Acronyms

GC Globular Cluster

PAndAS Pan-Andromeda Archaeological Survey

2MASS Two Micron All Sky Survey

NN Neural Network

RF Random Forest

MLP Multilayer Perceptron

RBC Revised Bologna Catalogue

L-BFGS Limited memory BFGS

ML Machine Learning

ROC Receiver Operating Characteristic

1 Introduction

A Globular Cluster (GC) is a gravitationally bound cluster of tens-of-thousands to millions of stars that occur in and around galaxies. They are objects of interest for research due to their properties that indicate stellar evolution and galaxy histories. Our galaxy, the Milky Way, contains over 150 known GCs, with more likely hiding behind the thick galactic disc. In contrast, M87, an enormous elliptical galaxy in the Virgo constellation, has been estimated to contain 12,000±800 GCs [39]. There is evidence to suggest some GCs form during major galaxy mergers [44, 9, 43] and in accretion events [27, 28]. They are some of the oldest objects within galaxies, with many having similar ages to their galaxy [24], and so can act as a fossil record with which to determine the histories of galaxy formation and evolution. Due to their high densities, close interactions of stars can occur which may give rise to exotic classes of stars such as blue stragglers[26]. This makes GCs useful for studying exotic physics that occurs less frequently in other parts of the universe. Globular clusters are generally found to have a bimodal colour distribution [9, 22, 23], forming two distinct populations. These populations are formed under different conditions and are still an area of research. GCs have varying metallicities (abundance of elements heavier than hydrogen and helium) and tend to consist mainly of population II stars [23]; the proportions of metals can be an indicator of a star's age, and also of the conditions in which they formed.

Identification of globular clusters is traditionally done via visual inspection of telescope data. Measurements from ground-based telescopes identify light sources that are categorised based on characteristics such as colour-magnitude and shape, and sources that match known characteristics of GCs are selected for visual inspection by astronomers to obtain a set of candidates. Although this method can be very accurate, this is a time-consuming process and requires skill and experience to do. Additionally, it is only practical to perform this method on small datasets, but with the exponential increase in astronomical data available it is simply unrealistic to manually check. Once candidates are selected, telescope time must be dedicated to making closer observations and gathering more accurate spectroscopic data. This is expensive and time-consuming and cannot be performed on the many thousands of possible candidates.

For this reason, it is necessary to build techniques for automating the process of narrowing down candidate lists. With the steady increase in computing power, Machine Learning (ML) techniques have become more widely accessible and can be used to perform a lot of the human work in a much smaller timescale. These techniques have been employed by many successfully, such as Barbisan et. al [5] who used random forest and neural network classifiers to identify GCs from ground based photometry, as well as for classification of galaxies by morphological features [12, 32, 34]. While not as accurate as spectroscopy, ML models are quick and easy to use and can still generate useful candidate lists. Furthermore, they require relatively little experience and knowledge to configure in comparison to the expertise required for traditional methods and spectroscopic analysis.

Identifying extragalactic GCs in distant galaxies is troublesome due to their appearance as single point sources, making them easily mistaken for background galaxies or foreground stars. One such solution to this issue is to combine ground-based photometry with high spatial-resolution imaging from the Hubble Space Telescope (HST) (e.g. [25]). High resolution imaging from HST has benefited GC research due to the ease at which candidates can be identified, even in distant galaxies. However, HST observations often point at the centre of galaxies and so do not include the wide halo of such galaxies, where many GCs lie. This causes samples to be incomplete as they do not include the GCs in the farther reaches of galaxies, and the GCs that are captured can be obscured by the galaxy's disc and surrounding dust and stars.

A focus of interest in GC research has been on the M31 galaxy. As M31 is our closest large galactic neighbour, and as a massive spiral galaxy, it provides an excellent comparison to our own Milky Way. Furthermore, due to it's close proximity (~780kpc [19]) detailed investigation of its system of globular clusters with space and ground-based telescope has been possible such as in [14, 36, 6, 17]. M31's was the first extragalactic globular cluster system to be studied [16], and since Hubble's identification of 140 possible clusters, many more have been added to the list by a variety of authors (e.g. [37, 29, 4, 7]).

In this paper we train two supervised ML classifiers on ground-based photometry data to identify potential GC sources, and use clustering techniques on the results to build a list of candidates. We train the models using a set of human confirmed sources matched onto a large survey of the area around and including the M31 & M33 galaxies, in two magnitude bands (g and i). Our aim is to create classifiers that can make these predictions, and to compile a set of potential GC candidates in the M31 halo.

In Section 2 we specify the project in greater depth (Section 2.1) and detail the aims and objectives. Section 3 outlines the data sets used (Section 3.1) and the methods of manipulating the data and applying it to the ML models (Section 3.2). In Section 4 we go into detail of the ML models used and their parameter tuning processes. We present the results in Section 5, including the comparison of the ML models' performances, and the application of the models to the wider data set. Finally we discuss the findings and the advantages and limitations of this approach in Section 6, and present the conclusions of the research.

2

Project Specification

2.1 Specification

This project aims to answer the questions:

- Can we generate useful globular cluster candidate lists for M31 using machine learning models trained on ground-based spectroscopic data?
- Can we still gain good results given the small training set?
- What are the limitations and benefits of the methods employed?

To answer these questions, we will implement machine learning algorithms in Python, and read in astronomical catalogue tables to gather data into a training set to fit the ML models. Multiple machine learning algorithms will be utilised in order to compare performance, and to ensure that a working model can be used to generate a suitable candidate list. It is clear from other studies that it is indeed possible to detect globular clusters with machine learning techniques. To assess if this work is successful we will use standard metrics to measure ML performance such as accuracy and recall, and area under the Receiver Operating Characteristic (ROC) curve. Recall will be the most important metric to focus on as we want to minimise the number of false negatives made by the models. As there are a relatively small number of GCs compared to other data points, we want to avoid missing as many as possible. Having a high number of false positives is not such an issue as they can be easily and quickly removed upon visual inspection.

Studies such as [11, 10] show that classification on datasets of similar length to that used here (2000 entries) is possible, however these studies used higher dimensional data, with up to 15 features.

Our expectations were that some useful predictions would be possible, but would not be accurate enough to solely rely on to generate candidate lists without extra human intervention.

3

Project Data & Methods

3.1 Data

The primary dataset used in this work is from the Pan-Andromeda Archaeological Survey (PAndAS) [30]: a large survey of the objects and structures around the Andromeda (M31) and Triangulum (M33) galaxies, covering an area of >400 square degrees. The survey, taken on the Canada–France–Hawaii Telescope, utilised the MegaPrime/MegaCam wide-field camera to obtain a detailed view of this area of the sky. This camera is comprised of 36 CCDs. Each pointing provided a usable view area of 0.96x0.94 deg^2, with each image making up one of the 406 individual fields that spans the area. Large catalogues are provided for each field, with each catalogue containing anywhere between 100,000 and 1,000,000 detected objects. To provide adequate colour discrimination of red-giant branch stars, PAndAS used g- and i-band filters in order to capture two distinct sections of the electro-magnetic spectrum. Excellent image quality is provided, with median image quality values of about 0.6 arcseconds for each band. Within the provided catalogues are columns indicating each object's location and photometry data in the two bands, along with their uncertainties. The data was sent for further image processing and photometric measurements using the Cambridge Astronomical Survey Unit (CASU) pipeline [20], which also provided an estimate of the object's classification as a stellar or non-stellar object, or just as noise/saturation.

In order to train an accurate ML classifier to identify GC candidates, we require a large set of accurately labelled samples. To provide this, alongside PAndAS, a master catalogue provided by Dr. Huxor, composed of version 5 of the Revised Bologna Catalogue (RBC) [13], among others [42, 41, 18], were used to construct the training data set.

The RBC catalogues objects such as GCs, stars, and galaxies in a region spanning 3 degrees around the centre of M31. This catalogue contains 2641 unique objects, totalling 730 confirmed GCs, 871 confirmed galaxies, 746 confirmed stars, and 13 extended clusters. Additionally, the catalogue provides 232 GC candidates, however these were not used in this work due to the

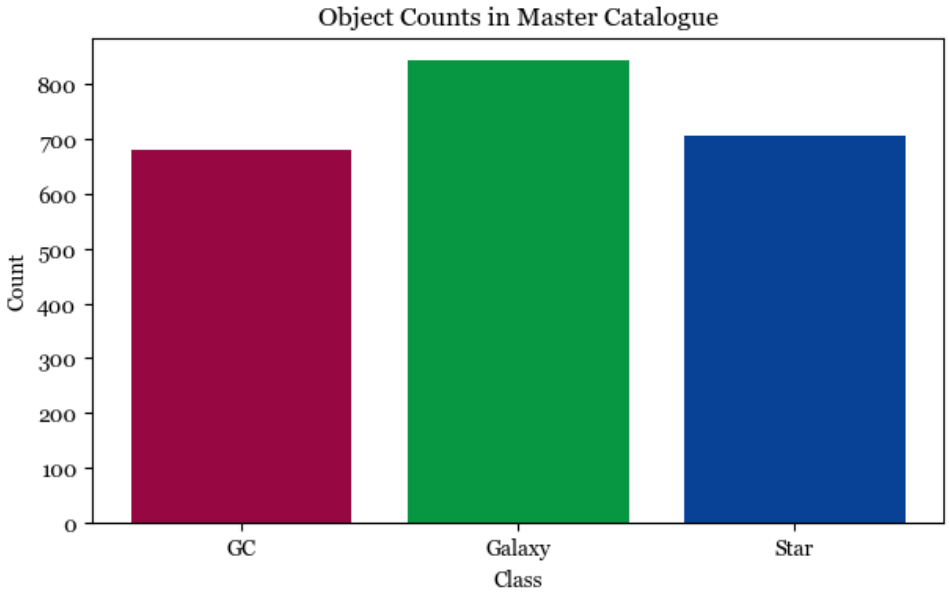

Figure 3.1: Object counts of the three main classes in the master catalogue.

uncertainty of their nature and so as to not risk training the algorithms with incorrect data. Finally there were 49 other objects that were not considered here, such as H-II regions and controversial objects. A bar chart of the three main classes are shown in Figure 3.1.

To generate the training data, the master catalogue was first cross-matched to the PAndAS catalogues to find all objects within an 18.0 arcsec radius of the location provided in the former table. Where the single closest match exceeded 1.0 arcsec from the confirmed source location, the object was removed from the training set as this indicated that either: the object was not picked up by PAndAS; or that the location provided in the master catalogue was inaccurate. Additionally, where the error margin on either of the magnitude values was greater than 0.05, the object was removed from training, so as to ensure that only the most accurate values are used to fit the ML models. The rest of the matches were saved to a table with the g and i magnitudes of the closest match added alongside the objects position, its given class (GC/extended cluster, galaxy, star), the error margin on the magnitude measurement, the number of objects found within 18.0 arcsec of the given location, as well as metadata for each point including the PAndAS field it lies within and its index in the original catalogue. The reason for measuring the object count within the specified radius is to determine if the object is crowded; if too many matches are found in a small area then the magnitude reading may be inaccurate due to overlap and blooming. Colour magnitude *g-i* was also added to the training set to provide an extra feature for learning.

Additional magnitudes were added from the Two Micron All Sky Survey (2MASS) [38]. 2MASS collected data between 1997-2001, imaging in the J, H, and K bands. Data from this survey was added to the training set in the hope of providing more features for the ML algorithms to learn from and therefore increase model accuracy. Objects in the training set were cross-matched with detected points in the 2MASS data in order to gather the respective mag-

nitudes in the J, H and K bands. The 2MASS data includes a column describing the quality of the observation, rated from A (best quality) down to F. Only results marked with A, B, or C quality were selected, implying a signal to noise ratio ≥5 (≥10 for A quality sources). Many of the objects (GCs and galaxies included) in the training set were not present in the reduced 2MASS data and so had to be excluded. The new second training set was reduced in size by almost half, with 381 galaxies and 344 GCs available.

3.2 Methods

All data handling and manipulation, and machine learning implementation was done in Python 3.9 using standard and specialised libraries, including pandas [40, 31] and numpy [15].

There are many ways to implement machine learning with a range of different algorithms available. For this research we used two different machine learning classifiers: Random Forest (RF) and Multilayer Perceptron (MLP), which is a type of Neural Network (NN). These algorithms were implemented using the scikit-learn library [35]. At first, multi-class classification was considered so that the models would differentiate between GCs, galaxies, and stars. It quickly became clear that the multi-class approach would not work as the models provided accuracy scores of around 50%, effectively being no better than a random classifier. Instead, a binary classification approach was much more effective, differentiating between GCs and non-GCs. Initially, stars were included in the training set, however the combination of stars and galaxies in a single 'non-GC' class appeared to confuse the classifier and it was more successful to remove them and to only have GCs and galaxies in the training set. Misclassifying stars did not prove to be an issue due to the ease that they can be ignored when viewing the results and later clustering techniques that would mostly eliminate them.

Tuning the ML models was done by measuring the model scores across different parameter values. Recall and accuracy scores were recorded on each iteration which could then be compared to evaluate which configuration gave the best performance of the model. Details of the tuning is explained in the relevant sections in Chapter 4.

For importing the large catalogue files, Astropy 5.1 was used [1, 2, 3]. Astropy is a Python package that provides utilities for astronomers and astrophysicist, supplying a wide range of tools that come in very useful in these fields. In this work, we used Astropy for catalogue file management, as well as its functions for matching object locations between catalogues which was used in generating the training data by matching the known objects in Revised Bologna Catalogue (RBC) against the PAndAS data, in order to gather colour-magnitude values for each object. This was also applied to 2MASS data for gathering magnitude values in other filter bands.

Outputs of the ML algorithm consisted of a classification on each relevant point in the PAndAS catalogue. For each field, up to 10,000 individual data points were identified as GCs by the ML model. Of course, this is highly impractical for the manual checking that's required to confirm the nature of an object as a potential candidate. It was noted, however, that where extended objects appeared in the images, such as large galaxies or globular clusters with

individually resolved stars, the model would identify most of the sources in the same manner; this meant that

For each run of the algorithm, a set of 5000-10000 GC predictions are made, which is impractical to manually look at. However, where there was an extended object such as a galaxy or cluster (when looking at only GC identifications), this showed up as a tight cluster of identifications as shown in figure X (screenshot of DS9 cluster of predictions). These tight clusters of points were indicative of objects of interest and so to collate these into single locations for manual analysis, DBSCAN was used. The benefit of using DBSCAN to identify these clusters of points is that any misclassifications of small individual points, such as stars or distant galaxies, are filtered out due to not being clustered. Each field then contained only a handful of points to manually check, making the search much more effective.

In case of slow execution of the machine learning models, the Hummingbird library [33] was considered for use. Hummingbird utilises the GPU in order to execute machine learning models at potentially higher speeds to the default CPU execution. This was not necessary after it became clear that the classifier speed was not a bottleneck, since it can execute in less than one second.

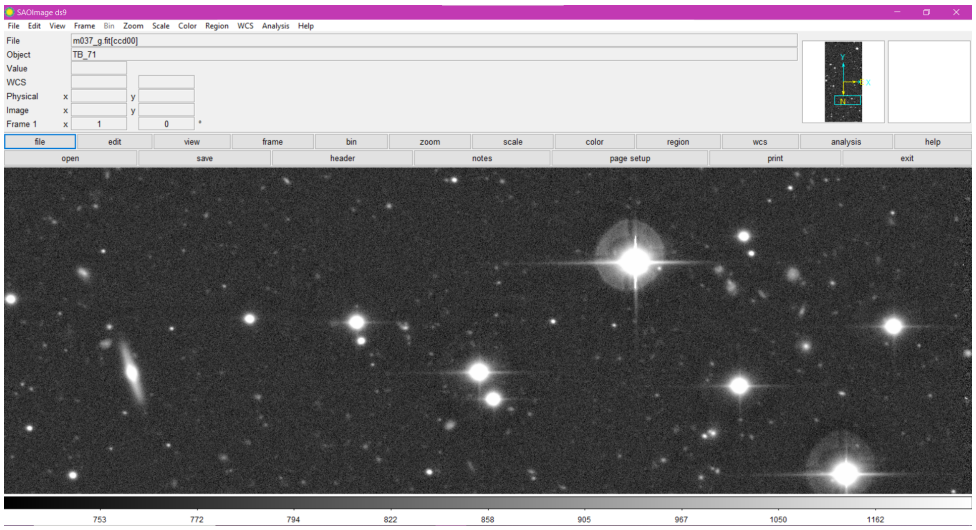

Figure 3.2: Screen-capture of the SAOImage DS9 software, viewing a field from the PAndAS data set.

It was of great importance to be able to visually analyse the objects we're studying, rather than only looking at the raw data. For this reason, we used the SAOImageDS9 software [21] to view the .FITS files provided by the PAndAS project. DS9 provides tools for viewing astronomical imaging data and creating regions on these images for easy analysis and object identification. This software was important for exploring the fields around M31 and understanding and analysing the output of the ML algorithms. A screenshot of the software in use on one of the fields is shown in Figure 3.2.

4 Models

In this chapter we describe the ML models and their parameter tuning. The application of these models to the data set for finding GC candidates is described in Chapter 5. Both models were tuned with the goal to maximise their recall values, so as to minimise the number of false negatives, this reasoning is described more in Section 2.1.

Table 4.1: Hyperparameters of the final RF model.

Parameter	**Value**
Criterion	*Gini*
Trees	*40*
Min. Samples Split	*0.5% of train samples*
Min. Leaf Samples	*2*
Max. Depth	*4*
Max. Features	*2*

Table 4.2: Hyperparameters of the final MLP model.

Parameter	**Value**
Activation Func.	*tanh*
Solver	*L-BFGS*
Hidden Layers Config.	*1 layer, 2 nodes*
alpha	*1.1*

4.1 Random Forest

Random forest is an algorithm that builds a set of tree-like structures to determine how to classify an input. Similar to a flowchart, the decision trees branch off at 'nodes' and use basic equality measures to determine which branch to send an input down. At the end of the branches are leaves that hold the classification possibilities, or labels, and act as the output of the decision

tree. RF uses an ensemble of these trees and takes the average or majority of the predictions to determine to overall output. This ensemble method yields a more accurate estimate and can prevent overfitting, which can be a problem on individual decision trees. The algorithm implements the bagging method for the ensemble, which was introduced by Breiman (1996) [8].

RF benefits from a reduced risk of overfitting due to the use of a large ensemble of trees. The averaging of many trees lowers the overall variance.

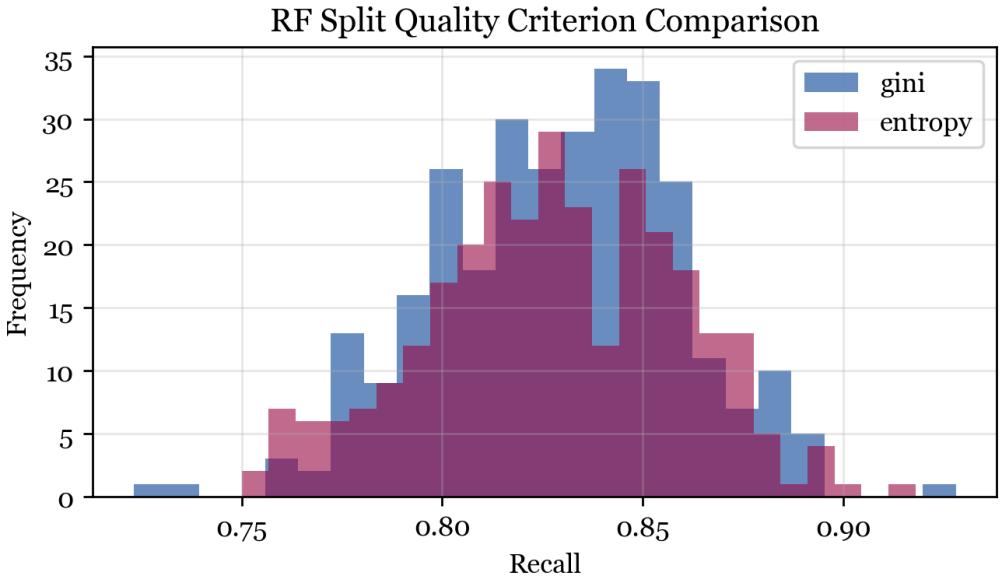

Figure 4.1: Histogram showing the results of 250 iterations of the random forest classifier's recall test scores for both 'gini' and 'entropy' criteria, trained using sklearn's default parameters.

Tuning of the RF algorithm required consideration of six different parameters: criterion for measuring the quality of a split, minimum samples required at each leaf node, number of trees in the ensemble, minimum samples to split a node, maximum features considered at each node, and maximum depth of the tree. The latter four parameters were tuned by performing a grid search on a range of values and performing 10-fold cross validation to determine which configuration of hyperparameters provided the best performance.

Split quality criterion was the first parameter to be chosen, with a comparison shown in Figure 4.1. The Gini impurity criterion provides a marginally better result, with slightly higher median and less variance in score, so was chosen as the parameter.

Final hyperparameter values for the RF model are shown in Table 4.1.

4.2 Multilayer Perceptron

Multilayer perceptron is a type of neural network that consists of a series of fully connected layers: an input and an output layer, as well as at least one hidden layer. Each node in the hidden layer is a neuron that uses an activation function to determine the strength of its output. The

network learns The weights of the connections between neurons are learnt via backpropagation.

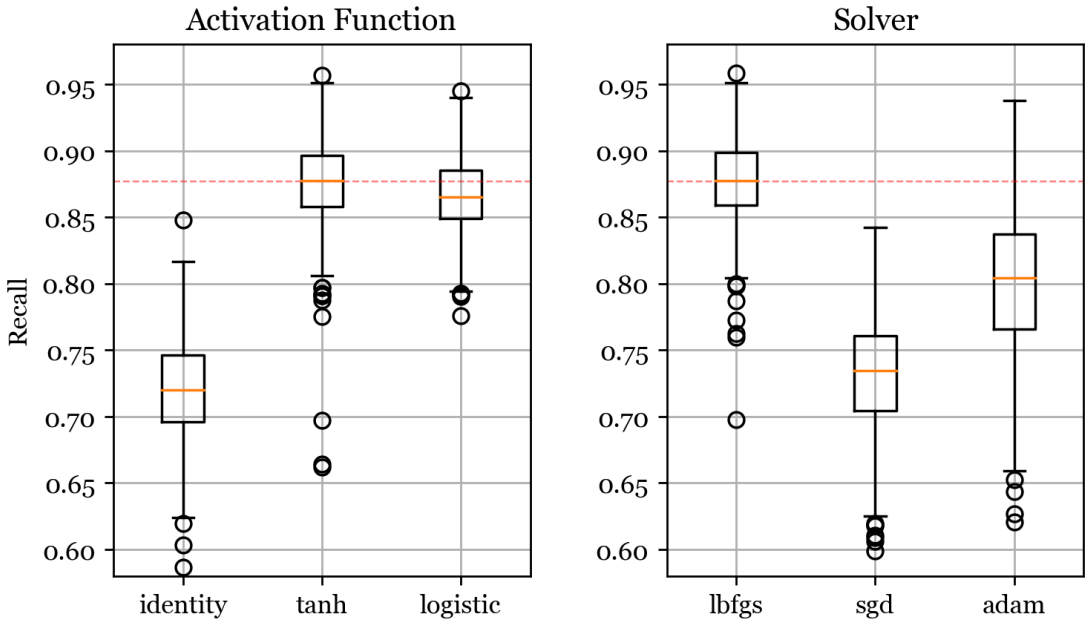

Figure 4.2: Box plots of the recall values after 250 iterations of the MLP classifier's test results. Left: Results from the different activation functions, *tanh* giving a marginally higher median score over logistic. Right: Results from the different solver algorithms, *lbfgs* giving considerably better performance over the other two solvers. The red dashed line at 0.877 represents the median score of the the best classifiers.

Four parameters were considered in tuning the MLP model, these were: the activation function, the solver algorithm for weight optimisation, the hidden layer size(s), and the L2 regularisation term (*alpha*). The final values for the MLP hyperparameters are shown in Table 4.2.

The first parameter considered was the activation function. Due to the nature of the data, the default activation function the rectified linear unit (ReLU) was unsuitable. ReLU returns the same input value if it's greater than zero, otherwise it returns zero. Our data is scaled and centered around zero, so using ReLU would result in half of the training data being ignored. For this reason, we consider the following activation functions: identity ($f(x) = x$), logistic function ($1/1 + exp(-x)$), and the hyperbolic tangent function ($f(x) = tanh(x)$) (referred to simply as *tanh* from here onwards). The results of 500 iterations, using an initial value *alpha*=1, the lbfgs solver, and a single hidden layer of two nodes, are shown in Figure 4.2 (left).

Following the results of these tests, *tanh* was chosen as the activation function as it performed almost equally as well as logistic, and as it is a symmetric function it applies better to the data being used here.

After the activation function was chosen, the solver was selected. Results of tests similar to the ones for the activation function are shown in Figure 4.2 (right). Stochastic gradient descent performed the worst, with the 'lbfgs' solver performing considerably better out of the three. Limited memory BFGS (L-BFGS) is an optimiser in the family of quasi-Newton methods. Adam

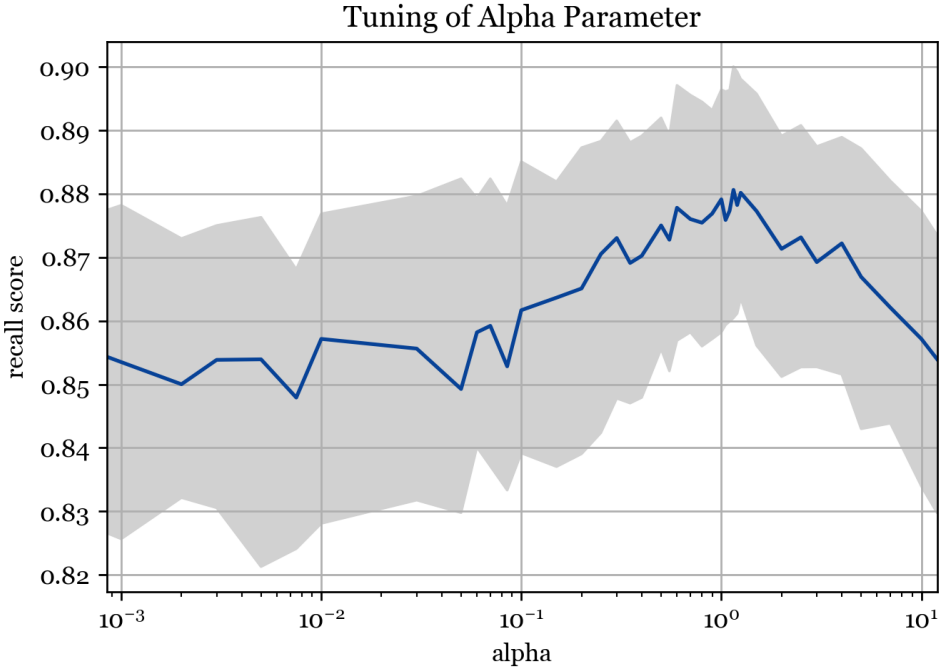

Figure 4.3: Tuning of the *alpha* parameter for the MLP model, with recall on the y-axis. Using the *lbfgs* solver, a range of *alpha* values were tested. The upper and lower boundaries of the grey area indicate the upper and lower quartiles of the test scores, with the blue line showing the median. An optimum score can be seen at around *alpha* = 1.1.

is an optimiser that is based on stochastic gradient descent. [expand on both of these]. Adam is better suited to large datasets, and also takes considerably longer to execute, taking an average of 916ms to train the classifier, whereas L-BFGS takes an average of only 33.7ms. The Adam optimiser may have the ability to perform as well as L-BFGS with parameter tuning, however due to the significantly longer training time L-BFGS clearly is the preferred choice. Additionally, L-BFGS only requires a single parameter, *alpha*, to be chosen to impact its classification ability which makes the parameter tuning process much easier.

Next, the hidden layer configuration was chosen. Selecting the configuration of the internal layers of a multilayer perceptron is done on a problem specific basis. Often, the number of hidden neurons is chosen to be in between the size of the input and the output layers, or to be twice the size of the input layer. Here, we tested various hidden layer configurations, using *tanh* activation and the L-BFGS solver. First, a rough optimisation of the *alpha* parameter was performed by training the classifier 100 times each on varying values of *alpha* and selecting the best value for each hidden layer configuration. Then, 250 MLPs were trained for each configuration and the distribution of their recall scores was compared. It was expected that just two neurons in a single hidden layer would be enough for the model to learn the data and this was confirmed by the experiments.

Finally, the *alpha* parameter was finely tuned for the selected configuration to reach the optimal configuration for this training set. Results of the *alpha* tuning is shown in Figure 4.3. A value of alpha=1.1 was chosen.

5 Results

In this section we will present the results of the work, including details on the different model's performances, as well as GC candidates found by the models.

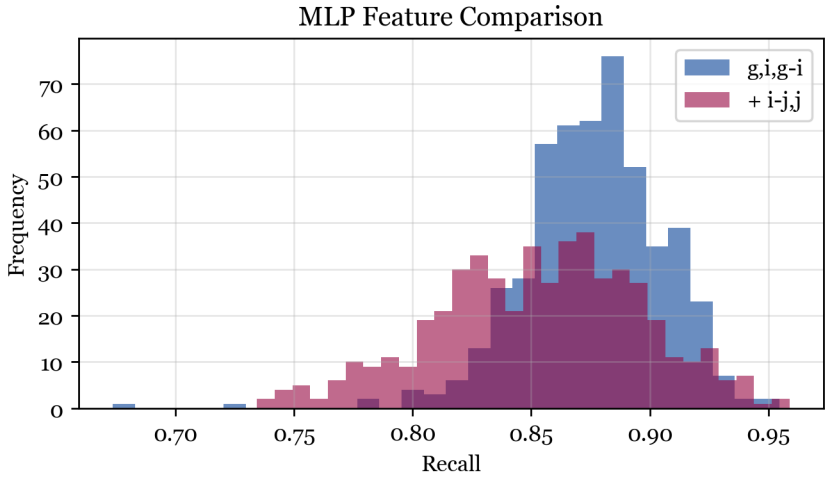

Figure 5.1: Comparison of recall values on two MLP models trained 500 times each with different subsets of the training data set.

5.1 Addition of 2MASS Data

Unfortunately, the observations from PAndAS could not be reliably cross-matched with those from 2MASS, due to the lower spatial resolution of 2MASS, and so the model could not be usefully applied for finding GC candidates across the field. Nonetheless, it was worth testing for a performance increase as a proof-of-concept. Figure 5.1 shows the spread of values on two MLP models trained 500 times each using different features. The model with the extra features used four nodes in the hidden layer and alpha=1. It's clear that that addition of the J and i-J features did not improve the overall performance of the model; using the initial three features

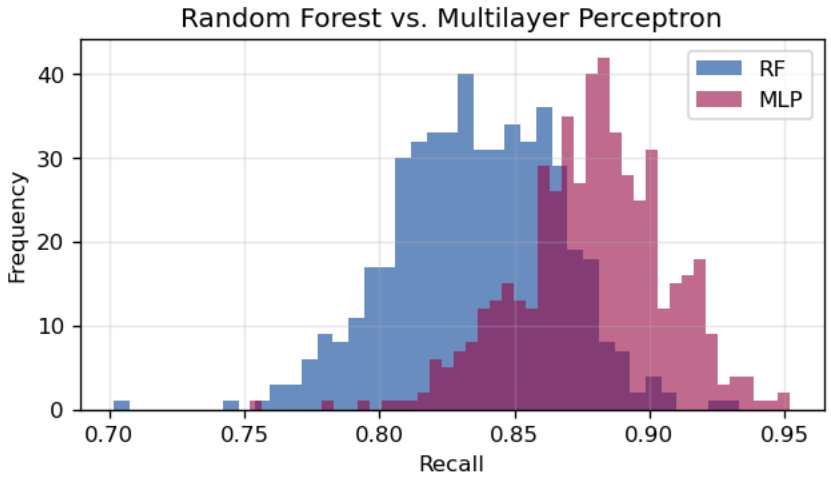

Figure 5.2: A comparison of the recall test scores of the RF model against the MLP model. 500 classifiers were trained and tested with 10-fold cross validation.

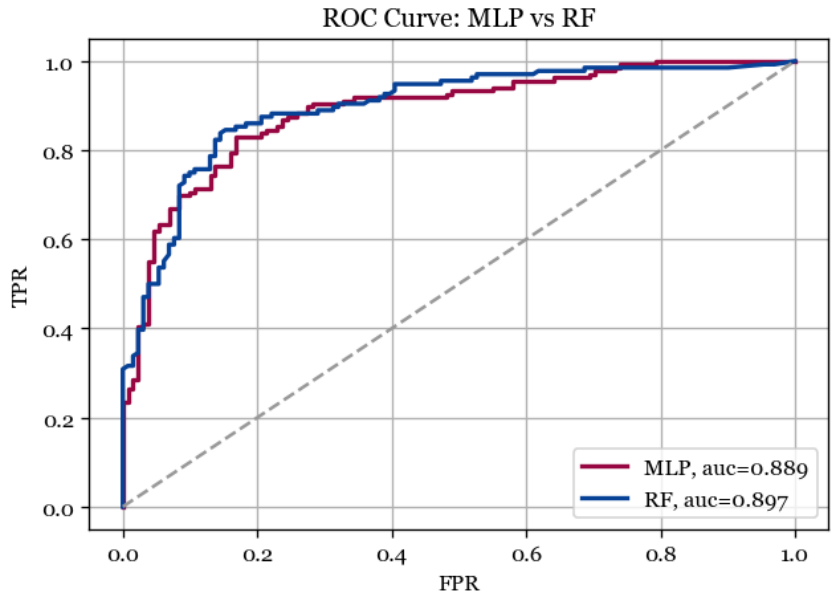

Figure 5.3: ROC curves for the two ML models, with area under the curve shown in the legend. The grey dotted line represents a random classifier.

gained a higher median score as well as a smaller spread of values. This is likely due to the fact that the addition of the 2MASS data necessarily reduced the size of the training set. The 2MASS data has a lower spatial resolution than PAndAS, so many of the GCs and galaxies from the training set simply had no corresponding data points in the 2MASS data and had to be excluded. For these reasons, the training set with J, H, and K, filter bands was not used in the final predictions.

5.2 Model Comparison

Both the random forest and multilayer perceptron classifiers performed well at classifying on the test data sets. Figure 5.2 shows a comparison of the recall scores of the two models, trained 500 times each and tested with 10-fold cross validation to gather a range of recall scores. We see here that MLP appears to perform best, with a median recall score of 88.0% and median accuracy of 82.7%, whereas the RF classifier has a median recall score of 83.6% and median accuracy of 83.5%. The ROC curves for both classifiers are shown in Figure 5.3, which tell that RF performs marginally better than MLP, with an area under the curve of 0.897, whereas MLP has an area under the curve of 0.889. This small difference may not be statistically significant.

5.3 DBSCAN Results & Potential Candidates

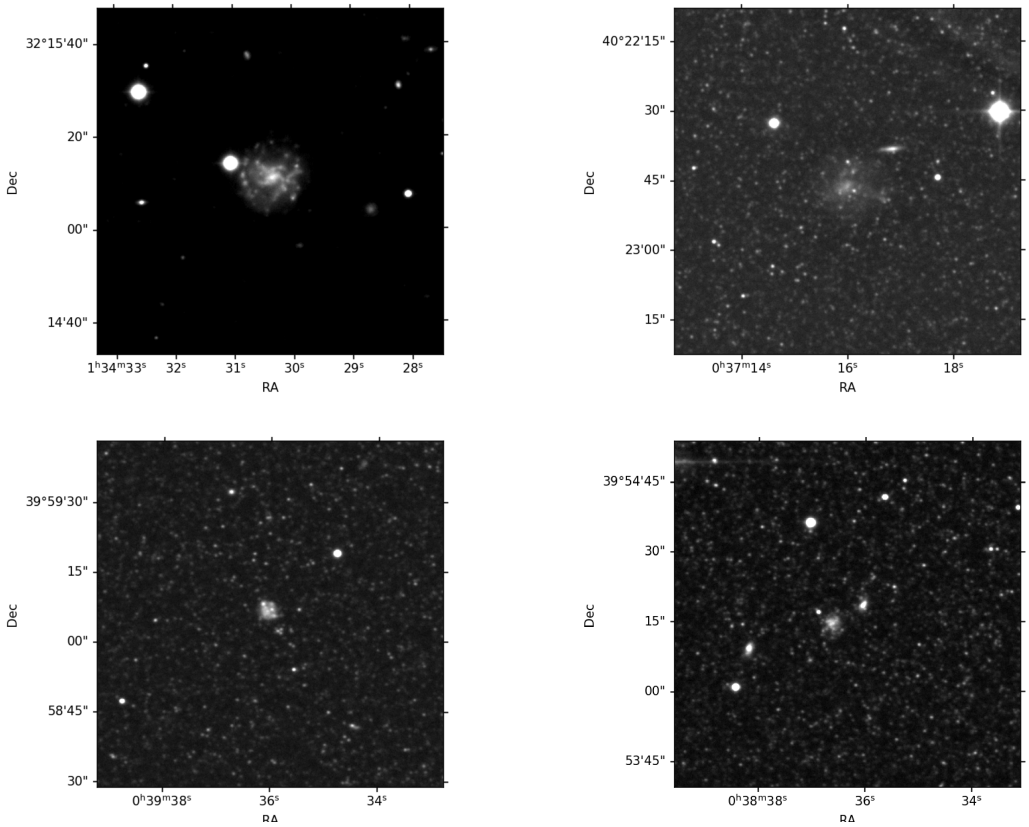

Figure 5.4: A set of four candidates found by MLP+DBSCAN.

The clustering using DBSCAN was successful in narrowing down the candidate lists. Most fields resulted in many thousands of GC classifications of which the majority were, of course, misclassifications. DBSCAN helped to solve this issue by only selecting the predictions that occurred in groups of four or more. This filtered out all of the individual predictions which were mostly background galaxies or foreground stars. The clusters left over consisted of large

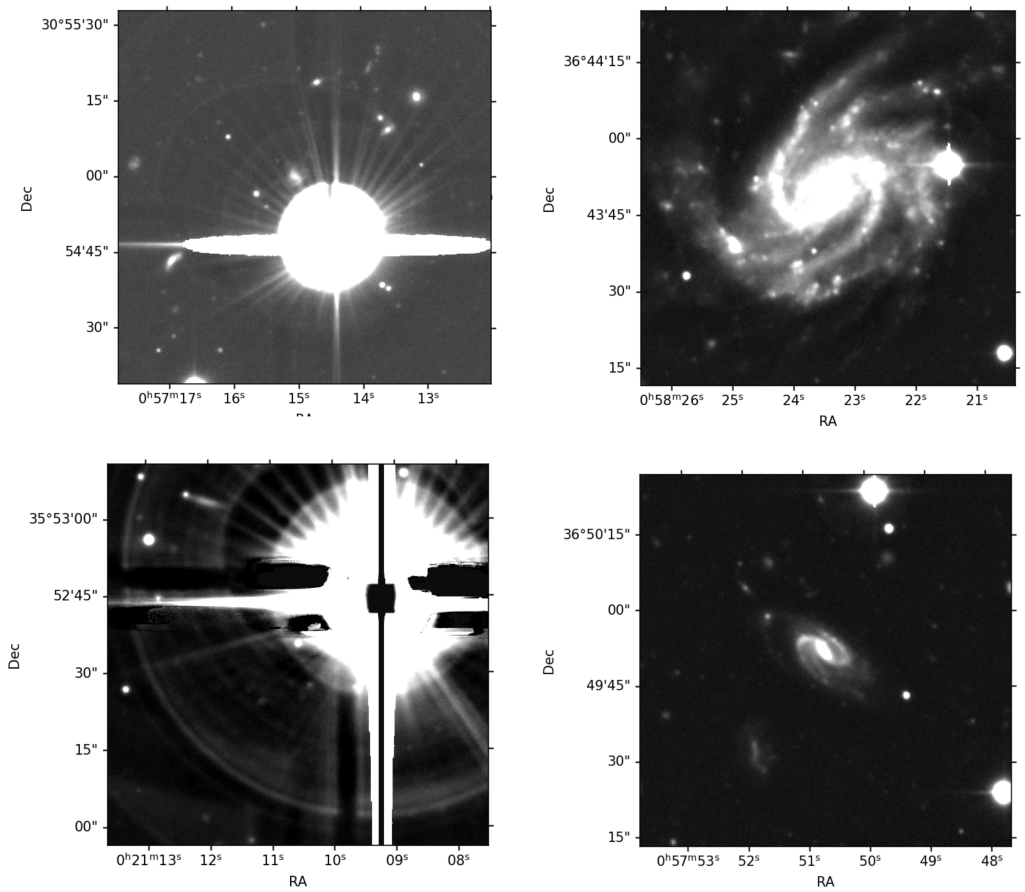

Figure 5.5: An example of four misclassifications made by MLP+DBSCAN.

galaxies, stars with diffraction spikes, and occasionally globular clusters. In Figure 5.4 we present four candidates for new GCs in the M31 halo, found by applying DBSCAN to results from the MLP classifier over a range of fields in the PAndAS dataset.

DBSCAN does also pick out many incorrectly classified objects as shown in Figure 5.5, where on the left we have two stars that were wrongly identified, as well as two galaxies on the right. The diffraction spikes from the stars show up as many individual sources in the PAndAS data and get misclassified by the ML model. Since all the sources are close together, they are identified as a relevant cluster by DBSCAN. Thankfully, all of these example are very easily ignored after visual inspection, so do not pose much of an issue for the manual checking phase after the candidates are produced.

6

Conclusion

In this paper we presented an approach to identifying candidate globular clusters using machine learning algorithms combined with a clustering algorithm. Random forest and multilayer perceptron were trained on the same training data, made up of g and i filter bands from the Pan-Andromeda Archaeological Survey, cross matched with a master catalogue provided by Dr. Huxor. Both machine learning models successfully learned the data and gained good test scores against the training set. After applying the models to the full PAndAS data set, DBSCAN was used to identify clusters of points in the predictions. These clusters signified objects with an extended structure, such as galaxies and globular clusters. The use of DBSCAN helped to narrow down the otherwise enormous candidate list, and made it feasible to look through the vast fields in the search for globular clusters. Although many misclassifications were made, most of these are easily filtered out by even an untrained eye after visual inspection and don't damage the value of the resultant candidate lists.

We attempted to apply data from the Two Micron All-Sky Survey to provide more training features for the ML models to learn from, but this was unsuccessful as 2MASS has a lower spatial resolution than PAndAS, and it was not worth sacrificing the extra detail for more training features.

We have shown that only two filter bands are required to generate useful candidate lists to speed up the search for globular clusters, and these accessible machine learning and clustering techniques can be applied to large scale surveys.

References

[1] Astropy Collaboration et al. "Astropy: A community Python package for astronomy". In: *Astronomy & Astrophysics* 558, A33 (Oct. 2013), A33. DOI: 10.1051/0004-6361/201322068. arXiv: 1307.6212.

[2] Astropy Collaboration et al. "The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package". In: *The Astronomical Journal* 156.3, 123 (Sept. 2018), p. 123. DOI: 10.3847/1538-3881/aabc4f. arXiv: 1801.02634 [astro-ph.IM].

[3] Astropy Collaboration et al. "The Astropy Project: Sustaining and Growing a Community-oriented Open-source Project and the Latest Major Release (v5.0) of the Core Package". In: *The Astrophysical Journal* 935.2, 167 (Aug. 2022), p. 167. DOI: 10.3847/1538-4357/ac7c74. arXiv: 2206.14220 [astro-ph.IM].

[4] Auriere, M., Coupinot, G., and Hecquet, J. "New globular clusters in the bulge of M31". In: *Astronomy and Astrophysics* 256.1 (1992), pp. 95–103.

[5] Emilia Barbisan et al. "Using machine learning to identify extragalactic globular cluster candidates from ground-based photometric surveys of M87". In: *Monthly Notices of the Royal Astronomical Society* 514.1 (May 2022), pp. 943–956. ISSN: 0035-8711. DOI: 10.1093/mnras/stac1396. eprint: https://academic.oup.com/mnras/article-pdf/514/1/943/43984840/stac1396.pdf. URL: https://doi.org/10.1093/mnras/stac1396.

[6] P Barmby et al. "M31 Globular Clusters: Colors and Metallicities". In: *The Astronomical Journal* 119.2 (Feb. 2000), p. 727. DOI: 10.1086/301213. URL: https://dx.doi.org/10.1086/301213.

[7] Battistini, P. L. et al. "New Globular Cluster Candidates in the Inner Regions of M31 and the Projected Density Profile of the Cluster System". In: *Astronomy and Astrophysics* 272 (1993), p. 77.

[8] Leo Breiman. "Bagging predictors". In: *Machine learning* 24 (1996), pp. 123–140.

[9] Jean P. Brodie and Jay Strader. "Extragalactic Globular Clusters and galaxy formation". In: *Annual Review of Astronomy and Astrophysics* 44.1 (Sept. 2006), pp. 193–267. DOI: 10.1146/annurev.astro.44.051905.092441.

[10] Tapanapong Chuntama et al. “Classification of astronomical objects in the Galaxy m81 using machine learning techniques II. an application of clustering in data pre-processing”. In: *2021 18th international joint conference on computer science and software engineering (JCSSE)*. IEEE. 2021, pp. 1–6.

[11] Tapanapong Chuntama et al. “Multiclass classification of astronomical objects in the galaxy m81 using machine learning techniques”. In: *2020 24th International Computer Science and Engineering Conference (ICSEC)*. IEEE. 2020, pp. 1–6.

[12] Jorge De La Calleja and Olac Fuentes. “Machine learning and image analysis for Morphological Galaxy Classification”. In: *Monthly Notices of the Royal Astronomical Society* 349.1 (Mar. 2004), pp. 87–93. DOI: 10.1111/j.1365-2966.2004.07442.x.

[13] S. Galleti et al. “2MASS NIR photometry for 693 candidate globular clusters in M31 and the revised Bologna catalogue”. In: *Astronomy & Astrophysics* 416.3 (Mar. 2004), pp. 917–924. DOI: 10.1051/0004-6361:20035632.

[14] Galleti, S. et al. “An updated survey of globular clusters in M31 - III. A spectroscopic metallicity scale for the Revised Bologna Catalog”. In: *A&A* 508.3 (2009), pp. 1285–1299. DOI: 10.1051/0004-6361/200912583.

[15] Charles R. Harris et al. “Array programming with NumPy”. In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

[16] Edwin Hubble. “Nebulous objects in messier 31 provisionally identified as globular clusters.” In: *The Astrophysical Journal* 76 (1932), pp. 44–69.

[17] John P. Huchra, Stephen M. Kent, and Jean P. Brodie. “Extragalactic Globular Clusters. II - the M31 globular cluster system”. In: *The Astrophysical Journal* 370 (1991), pp. 495–504. DOI: 10.1086/169836.

[18] A Huxor et al. “The discovery of remote globular clusters in M33”. In: *The Astrophysical Journal* 698.2 (2009), p. L77.

[19] A. P. Huxor et al. “The outer halo globular cluster system of M31 – I. The final PAndAS catalogue”. In: *Monthly Notices of the Royal Astronomical Society* 442.3 (June 2014), pp. 2165–2187. ISSN: 0035-8711. DOI: 10.1093/mnras/stu771. eprint: https://academic.oup.com/mnras/article-pdf/442/3/2165/3540511/stu771.pdf. URL: https://doi.org/10.1093/mnras/stu771.

[20] Mike J. Irwin et al. “Vista data flow system: Pipeline processing for WFCAM and Vista”. In: *SPIE Proceedings* 5493 (Sept. 2004). DOI: 10.1117/12.551449.

[21] W. A. Joye and E. Mandel. “New Features of SAOImage DS9, Astronomical Data Analysis Software and Systems XII”. In: Astronomical Society of the Pacific Conference Series 295 (Jan. 2003). Ed. by H. E. Payne, R. I. Jedrzejewski, and R. N. Hook, p. 489.

[22] T. D. Kinman. "Globular Clusters, II. The Spectral Types of Individual Stars and of the Integrated Light". In: *Monthly Notices of the Royal Astronomical Society* 119.5 (Oct. 1959), pp. 538–558. ISSN: 0035-8711. DOI: 10.1093/mnras/119.5.538. eprint: https://academic.oup.com/mnras/article-pdf/119/5/538/8078093/mnras119-0538.pdf.

[23] Robert P. Kraft. "On the nonhomogeneity of metal abundances in stars of globular clusters and satellite subsystems of the galaxy". In: *Annual Review of Astronomy and Astrophysics* 17.1 (1979), pp. 309–343. DOI: 10.1146/annurev.aa.17.090179.001521.

[24] Lawrence M. Krauss and Brian Chaboyer. "Age Estimates of Globular Clusters in the Milky Way: Constraints on Cosmology". In: *Science* 299.5603 (2003), pp. 65–69. DOI: 10.1126/science.1075631. eprint: https://www.science.org/doi/pdf/10.1126/science.1075631. URL: https://www.science.org/doi/abs/10.1126/science.1075631.

[25] Arunav Kundu and Bradley C. Whitmore. "New Insights from HST Studies of Globular Cluster Systems. I. Colors, Distances, and Specific Frequencies of 28 Elliptical Galaxies". In: *The Astronomical Journal* 121.6 (June 2001), p. 2950. DOI: 10.1086/321073. URL: https://dx.doi.org/10.1086/321073.

[26] Peter J. T. Leonard. "Stellar Collisions in Globular Clusters and the Blue Straggler Problem". In: *The Astronomical Journal* 98 (1989), p. 217. DOI: 10.1086/115138.

[27] A. D. Mackey et al. "Evidence for an accretion origin for the outer halo globular cluster system of M31". In: *The Astrophysical Journal Letters* 717.1 (2010). DOI: 10.1088/2041-8205/717/1/l11.

[28] Dougal Mackey et al. "Two major accretion epochs in M31 from two distinct populations of globular clusters". In: *Nature* 574.7776 (Oct. 2019), pp. 69–71. DOI: 10.1038/s41586-019-1597-1.

[29] N. U. Mayall and O. J. Eggen. "Four nebulous objects in the outer parts of the Andromeda nebula". In: *Publications of the Astronomical Society of the Pacific* 65.382 (1953), pp. 24–29. ISSN: 00046280, 15383873. URL: http://www.jstor.org/stable/40675902.

[30] Alan W. McConnachie et al. "The remnants of Galaxy Formation from a panoramic survey of the region around M31". In: *Nature* 461.7260 (2009), pp. 66–69. DOI: 10.1038/nature08327.

[31] Wes McKinney. "Data Structures for Statistical Computing in Python". In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman. 2010, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

[32] A. Naim et al. "Automated morphological classification of APM galaxies by supervised Artificial Neural Networks". In: *Monthly Notices of the Royal Astronomical Society* 275.3 (Jan. 1995), pp. 567–590. DOI: 10.1093/mnras/275.3.567.

[33] Supun Nakandala et al. "Compiling classical ml pipelines into tensor computations for one-size-fits-all prediction serving". In: *Systems for ML workshop at NeurIPS*. 2019.

[34] E. A. Owens, R. E. Griffiths, and K. U. Ratnatunga. "Using oblique decision trees for the morphological classification of Galaxies". In: *Monthly Notices of the Royal Astronomical Society* 281.1 (July 1996), pp. 153–157. DOI: 10.1093/mnras/281.1.153.

[35] Fabian Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12.85 (2011), pp. 2825–2830. URL: http://jmlr.org/papers/v12/pedregosa11a.html.

[36] Soo-Chang Rey et al. "GALEX Ultraviolet Photometry of Globular Clusters in M31: Three-Year Results and a Catalog". In: *The Astrophysical Journal Supplement Series* 173.2 (Dec. 2007), p. 643. DOI: 10.1086/516649. URL: https://dx.doi.org/10.1086/516649.

[37] Seyfert, C. K. and Nassau, J. J. "Nebulous Objects in the Andromeda Nebula." In: *The Astrophysical Journal* 102 (1945), p. 377.

[38] M. F. Skrutskie et al. "The Two micron all sky survey (2MASS)". In: *The Astronomical Journal* 131.2 (2006), pp. 1163–1183. DOI: 10.1086/498708.

[39] Naoyuki Tamura et al. "A Subaru/Suprime-Cam wide-field survey of globular cluster populations around M87 – I. Observation, data analysis and luminosity function". In: *Monthly Notices of the Royal Astronomical Society* 373.2 (Oct. 2006), pp. 588–600. ISSN: 0035-8711. DOI: 10.1111/j.1365-2966.2006.11067.x. eprint: https://academic.oup.com/mnras/article-pdf/373/2/588/4101352/mnras0373-0588.pdf. URL: https://doi.org/10.1111/j.1365-2966.2006.11067.x.

[40] The pandas development team. *pandas-dev/pandas: Pandas*. Version latest. Feb. 2020. DOI: 10.5281/zenodo.3509134. URL: https://doi.org/10.5281/zenodo.3509134.

[41] Graziella di Tullio Zinn and Robert Zinn. "A search for intergalactic globular clusters in the local group". In: *The Astronomical Journal* 149.4 (2015), p. 139.

[42] Graziella di Tullio Zinn and Robert Zinn. "More remote globular clusters in the outer halo of M31". In: *The Astronomical Journal* 145.2 (2013), p. 50.

[43] Bradley C. Whitmore and Francois Schweizer. "Hubble Space Telescope observations of young star clusters in NGC-4038/4039, 'the antennae' galaxies". In: *The Astronomical Journal* 109.3 (1995), pp. 960–980. DOI: 10.1086/117334.

[44] S. E. Zepf and K. M. Ashman. "Globular cluster systems formed in galaxy mergers". In: *Monthly Notices of the Royal Astronomical Society* 264.3 (1993), pp. 611–618. DOI: 10.1093/mnras/264.3.611.

Acknowledgments