A COMPARATIVE STUDY OF MACHINE LEARNING ALGORITHMS APPLIED TO SPECTRAL STELLAR CLASSIFICATION WITH RESPECT TO TRAINING SAMPLE SIZE

Carl Wilson
School of Computing & Engineering
University of Huddersfield
Huddersfield, UK
u0370630@unimail.hud.ac.uk

***Abstract*— This paper compares the relative performance of conventional machine learning techniques against adaptive neuro-fuzzy interface system, with respect to training sample size, when applied to stellar classification. Through a carefully designed set of experiments it will be shown that conventional techniques such as Logistic Regression and Multi-Layer Perceptron perform remarkably well on ever diminishing training sample sizes.**

Keywords—Machine Learning, ANFIS, SVM, Logistic Regression, ANN, MLP, Stellar, Classification

I. INTRODUCTION

The accurate identification of stellar objects presents, potentially, one of the largest naturally occurring datasets in existence. This dataset, which is rapidly increasing through more sophisticated surveys, cannot be analysed with traditional techniques and so must turn to machine learning which can efficiently and accurately identify and classify stellar objects. This is an important pursuit as understanding our universe helps us understand many fundamentals of our planet and its neighbourhood.

II. LITERATURE REVIEW

A range of machine learning techniques and algorithms have historically been applied to the problem of spectral stellar classification, all with strong performance. K-Means clustering, as an unsupervised method, has been explored by (Garcia-Dias et al., 2018) on the Apache Point Observatory Galactic Evolution Experiment [APOGEE] dataset which is a portion of the Sloan Digital Sky Survey [SDSS] dataset. This work focussed on the near-infrared spectrum Similarly, Logistic Regression has been applied to stellar classification by (Beitia-Antero et al., 2018), using far and near ultraviolet light bands to classify different T Tauris Stars from background stellar objects. The resulting confusion matrix showed strong performance of 97.9% accuracy; however, it must be noted there was significant class imbalance between objects of interest and background stellar objects.

Random Forests [RFs] used to classify stars, galaxies and quasars by (Bai et al., 2018) achieved 99% accuracy for stars and galaxies and 94% for quasars. This work also went further, making use of the insights that RFs offer into the interactions between predictor variables, to understand the nature of each class. The study benchmarked different machine learning algorithms, such as Decision Tree, K-Nearest Neighbours [KNN], RF and Support Vector Machines [SVM] and found that Random Forests provided the highest accuracy for a very low computational overhead.

SVMs were explored in detail by (Krakowski et al., 2016) where 10-fold cross-validation with a grid-search of hyper-parameters [optimising regularisation and the kernel coefficient] to tune the model. The data was split into bins of light spectra and models fitted to each bin and the resulting outputs aggregated. This was completed for two sets of models, one with five parameters and one with six parameters with the resulting models scoring 91.8% and 97.3% respectively when considering accuracy [sum of true observations to sum of all observations]. SVMs have also been explored by (Kovács & Szapudi, 2015) in the instance of separating galaxies, quasars and stars. The focus of this study was on the infra-red and redshift predictors.

In the case of (Zhao et al., 2022) a Convolutional Neural Network [CNN] and Ensemble Convolutional Neural Network [ECNN] were applied to SDSS datasets to classify stellar objects where the signal to noise ratio was quite low. The ECNN worked by having multiple classifiers and using bagging to integrate a classification. A total of six shallow CNNs were bagged, with weightings tuned using validation data to develop a model which generalised well without high computational overhead. The Ensemble model outperformed all the sub-models producing a validation score of 95%. The ECNN was benchmarked against SVM and RF, both using Principal Component Analysis [PCA] for dimensionality reduction first and was found to outperform them. The key challenges present with SVMs and Artificial Neural Network [ANN], also called Multi-Layer Perceptron [MLP] architectures, is the lack of explain-ability due to the "black-box" nature of the algorithms mapping function.

Adaptive Neuro-Fuzzy Inference Systems [ANFIS] seemingly offer the non-linear mapping and learning ability of neural networks with the knowledge representation and explain-ability of fuzzy systems, two strengths that could not only potentially provide models capable of high levels of performance but also an ability to tap into the mapping between predictor variable and prediction to understand the relationship and further develop our knowledge. However, little evidence has been found of ANFIS being applied to this domain – while theory and other research suggests ANFIS can be a powerful algorithm for classification, particularly for fast learning from smaller training sample sizes.

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

In the instance of (Nagarathinam & Ponnuchamy, 2019) ANFIS was applied to brain tumour detection and segmentation and achieved 98.4% accuracy and 96.7% specificity on segmentation using the Brain Tumour Segmentation [BraTS] 2015 dataset. This was compared against 96.5% and 94.2% for accuracy and specificity for SVM classification models that had been completed previously.

ANFIS has previously been applied to detection and segmentation of cracks in weld images by (Mohana Sundari & Sivakumar, 2021) and when compared to CNN methods the ANFIS model achieved 97% specificity and 98% segmentation accuracy versus 95% and 96% for specificity and segmentation accuracy for the conventional CNN. It was noted that feature extraction from the images was crucial for the performance of the model, where images were enhanced for resolution.

In the instance of (Zhang et al., 2013) ANFIS was applied to simulated fault data for a power distribution system, where there were ten potential fault modes. The models were run with a significant number of epochs [10,000 as stopping condition] or the RMSE was smaller than 0.0003. As the output of the ANFIS is not an integer relating to the fault modes, the results were rounded to the nearest expected prediction. Once trained a test sample of 3600 cases were used to test the performance of the ANFIS models that had been built, including samples that were outside the original boundary of the training date. The accuracy rate was determined to be 99.4% which was significantly higher than the comparison case which used an ANN.

A comparison of multiple models, including ANN, ANFIS, SVM and RF when classifying heating values of burning municipal solid municipal waste was conducted by (You et al., 2017) and found that the precisions of the ANN, ANFIS, SVM and RF models were 73.5%, 94%, 87.5% and 90% respectively. It should be noted the back-propagation mode used for the ANN, only had 1 hidden layer which would severely hinder the ANN's performance versus an ANFIS model which has multiple levels of abstraction to map non-linear functions between input and out. While the ANN, SVM and RF models took very little time to train, the ANFIS model took significantly longer – but was still in the order of magnitude of seconds, rather than hours or days.

Finally, membership functions within ANFIS were investigated by (Talpur et al., 2017) with regards to different classification problems. Several standard and familiar benchmark datasets were examined, such as Iris, Teaching Assistant Evaluation and Breast Cancer – all having different numbers of features from 4 to 10 and numbers of observations from 150 to 1728. The ANFIS settings examined were the different types of membership function and number of membership functions used to build models to a fixed number of epochs. A hold-out validation portion of 0.3 was used for testing with the remaining 0.7 used for training. The type of membership functions included the default grid-partitioning that MatLab Fuzzy Logic Toolbox employs as default and will later be used in the study presented here with the SDSS dataset. It was found that Gaussian input functions were typically higher performing than generalised bell shape, trapezoidal and triangular. Interestingly Gaussian performed the worst on the largest dataset but was much better on the smaller training samples. It was observed that the algorithms, across all the datasets, did not appear to generalise well – however the performance metric used was RMSE, which makes it difficult to evaluate if the models were overfitted.

The bias-variance trade-off and managing over-fitting of machine learning algorithms is a key challenge faced by Data Scientists and Machine Learning Practitioners today. Finding an algorithm that makes a suitable assumption about the underlying characteristic can lead to a much better generalising model and lower computational overheads, whereas using algorithms that make little or no assumptions can potentially produce a map between predictor variables and predictions that does not actually exist. There is an appropriate limit between pushing a given algorithm to improve accuracy before starting to map noise. Somewhere between these is the optimum error, which only include irreducible error or noise.

While datasets for model training may contain sample sizes in the order of magnitude of 10^5 or even 10^6; this is a fraction of the estimated 10^{24} stars in the observable universe. By flipping the problem with a fixed dataset size – the training sample can be reduced as a ratio of the testing sample to understand which techniques and algorithms tend to generalise better for a much larger test sample relative to the training sample when classifying stellar objects using SSDS dataset.

This report will aim to benchmark conventional machine learning algorithms, such as Logistic Regression and SVM against ANFIS on a range of training sample sizes and measured on performance for predictive capability on a fixed blind test dataset.

III. Design of Experiments

A total of four conventional algorithms were selected based on the literature review to give a good overview of performance, they were: K-Nearest Neighbours [KNN], Logistic Regression, Support Vector Classification [SVC], Multi-Layer Perceptron [MLP]. Additionally, two Adaptive Neuro-Fuzzy Inference Systems [ANFIS] with 2 and 3 membership functions [ANFIS2 and ANFIS3 respectively] were also selected for the study.

The solutions were benchmarked on training and test accuracy and the F1 score, also known as the balanced F-score. An additional metric was devised to benchmark over-fitting, which was the delta between the training accuracy and the test accuracy.

The experiment was designed to challenge each algorithm on a progressively diminishing training sample size from 78,052 down to 78, but always testing on the same 21,947 observations in the test sample to demonstrate which algorithm is the most robust for generalising with smaller training samples, as can be observed in Table I.

TABLE I. Parameters for Design of Experiments

Set	Variables	Test Sample Size	Training Sample Size	Training Fraction [%]
1	6	21,947	78,052	78.05%
2	6	21,947	15,610	15.61%
3	6	21,947	7,805	7.81%
4	6	21,947	1,561	1.56%
5	6	21,947	781	0.78%
6	6	21,947	156	0.16%

Set	Variables	Test Sample Size	Training Sample Size	Training Fraction [%]
7	6	21,947	78	0.08%

IV. EXPLORATORY DATA ANALYSIS

The dataset used was taken from the Sloan Digital Sky Survey [SDSS] release 17 and featured 100,00 observations for 17 feature columns and 1 class column identifying each object as a galaxy, star or quasar (Fedesoriano, 2022).

The raw data was examined in Pandas, there were no missing values but there was a significant outlier which was removed. The six key predictor variables, across the color spectra, were renamed to something more meaningful. The dataset was found to be imbalanced between the target classes, driving the requirement for stratification in the down-sampling and suitable scoring metrics to handle imbalanced predictions against ground truth.

Just over 78% of all observations were of unique objects, hence this was used as the training and test split with all repeat observations, when ordered by modified Julian date, falling into holdout test sample.

Completing a distribution test against a Gaussian distribution where the null hypothesis was the distribution was Gaussian was found to be not true; thus, the data was not normally distributed for any of the predictor variables.

A pair-plot using Seaborn was used to examine the distribution of the different predictor variables with respect to class but also how they interacted with each other, see Fig. 1.

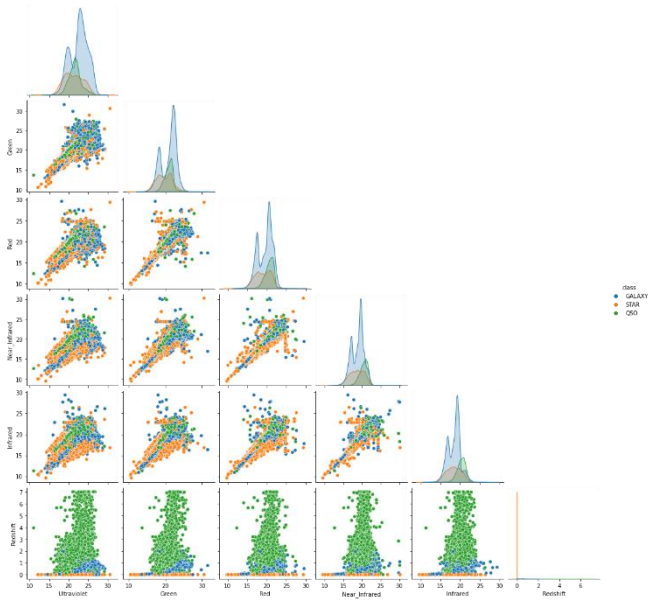

Fig. 1. Pair-plot for predictor variables against classes

It was evident that the Redshift variable, see Fig. 2, was crucial in identifying the QSO class from STAR and GALAXY, however much more subtle differences in the interactions between other variables are required to be mapped to predict STAR from GALAXY.

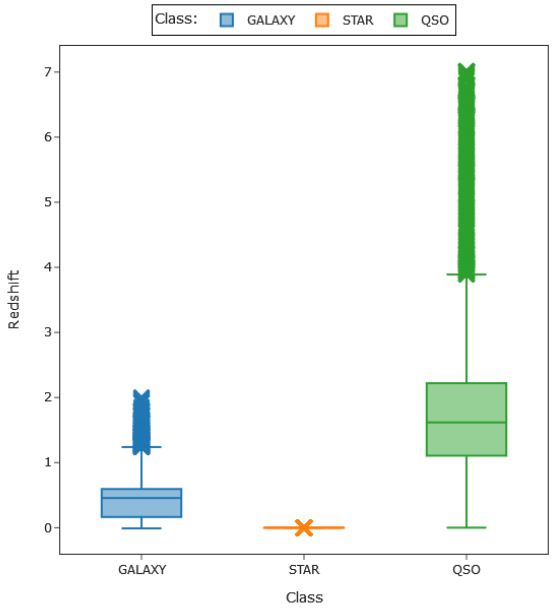

Fig. 2. Boxplot of Reshift for each Class.

V. IMPLEMENTATION

A. *Data Preparation*

Stratification was used to ensure the same proportion of each class label was included in all training set sizes and the test set size was fixed and remained constant for all model evaluations. The classes were encoded using an ordinal encoder and predictor variables were normalised using a min-max scaler due to the data not fitting a Gaussian distribution.

B. *Conventional Machine Learning Appliction*

All four conventional machine learning algorithms were applied using SciKit-Learn, using ten-fold stratified cross-validation with a validation fraction of 0.2 applied with grid-searching across various hyper-parameters for each respective algorithm. The hyper-parameters that were tuned can be observed in Table II.

Each model using each algorithm was optimised using a grid-search technique, with the optimum hyper-parameters being selected for the model that had the highest mean validation score.

TABLE II. HYPER-PARMETERS BY ALGORITHM

Algorithm	Hyper Parameters
KNN	Algorithm, Leaf Size
Logistic Regression	C – Regularisation
SVM	C – Regularisation, Kernel
MLP	Hidden Layer Sizes, Activation Functions, Solver, Learning Rate

C. ANFIS Application

ANFIS was implemented via MatLab and the models were trained on the same datasets and normalized predictor variables as the conventional machine learning algorithms used with fuzzification achieved through grid partitioning.

The fuzzy inference system was optimised using a hybrid method of backpropagation and least-squares estimation. A random, stratified, hold out sample of 0.2 was used for validation. An initial run with a much higher number of epochs was completed to purposely over-fit the model and understand the optimum number of epochs to achieve good generalisation. The results can be found in Table III.

TABLE III. ANFIS OPTIMUM EPOCH NUMBER BY SAMPLE SIZE

Training Sample Size	Optimum Epoch Number	
	ANFIS2	***ANFIS3***
78	30	29
156	35	13
781	31	15
1,561	39	17
7,805	19	21
15,610	21	N/A
78,052	24	N/A

The break-point was determined by examining the train and validation error versus epoch and identifying the point where there is a minima and the validation error becomes unstable and starts to significantly increase, see Fig. 3.

Once the optimum number of epochs was determined, the experiment was rerun with the being used to predict against the training and test datasets for posts-processing and results analysis.

Training sample sizes 15,610 and 78,052 were not explored with three membership functions due to the significant computational overhead, which limits the conclusions of this study.

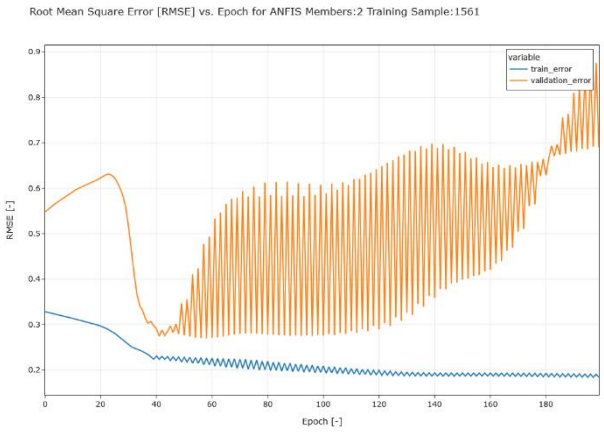

Fig. 3. RMSE vs. Epoch for ANFIS2 with 1,561 training sample.

D. Post Processing and Results Wrangling

The test samples for all algorithms and training sample sizes, were of the same length – allowing a single concatenated matrix of predictions. The training samples for all algorithms were grouped by training sample size and concatenated.

A simple rounding method was used to convert the raw ANFIS output floating point number into one of the three expected class integers.

VI. RESULTS

All metrics by algorithm and training sample size can be found in Table IV. The F1 scores, corrected for class imbalance, indicate that conventional machine learning algorithms were more consistent than the ANFIS algorithms, as can be observed in Fig. 4. MLP appeared to be the most consistent, barring one outlier with Logistic Regression and SVR also exhibiting high performance.

When considering the distribution of F1 Scores on the test sample, with the distribution of the training scores with the training sample sizes – it appears as though the ANFIS models are heavily over-fitted and not generalizing as well as the conventional algorithms as can be observed in Fig. 5.

The training scores indicated that all models had been trained to a reasonable degree, across all training set sizes with a few notable exceptions: ANFIS2 typically performed lower than the other algorithms for the larger training set sizes and MLP and KNN both performed well with one outlier each.

Examining the F1 scores versus training set size, in Fig.6, it can be clearly observed the marked difference between ANFIS and the conventional algorithms and just how consistent Logistic Regression and SVC were. This overfitting becomes more apparent when examining a scatter plot of testing score versus training score, see Fig. 7.

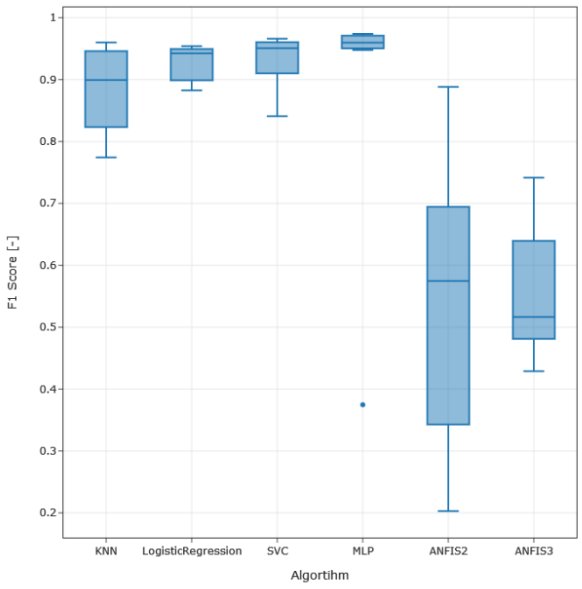

Fig. 4. Boxplot of F1 scores by algorithm for all training samples.

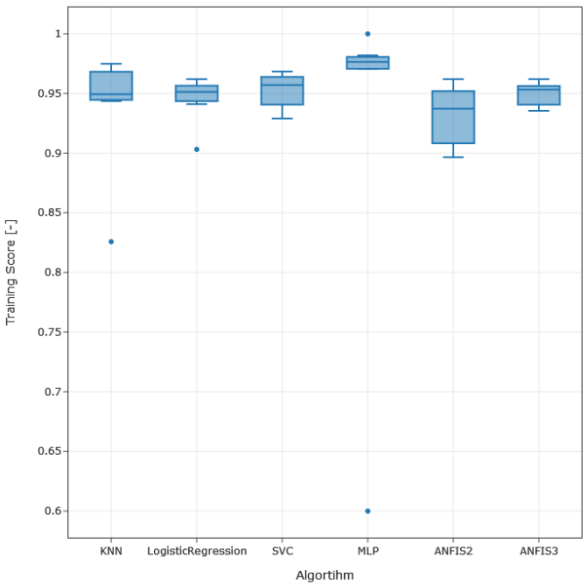

Fig. 5. Boxplot of training scores by algorithm for all training samples.

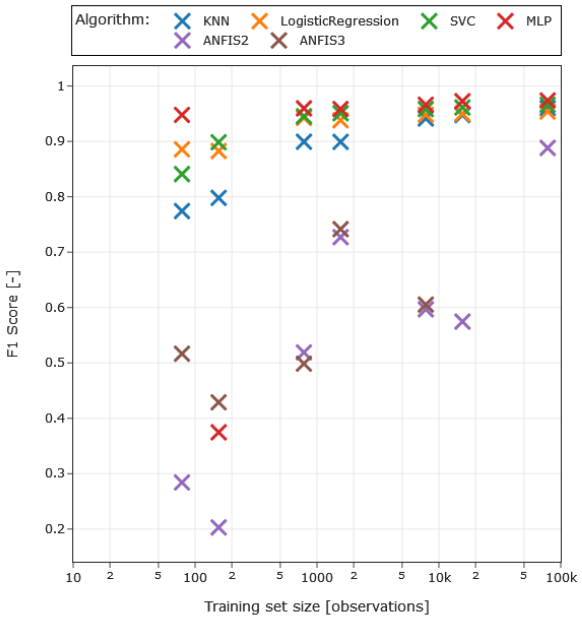

Fig. 6. Scatter plot of F1 score against training sample size by algorithm.

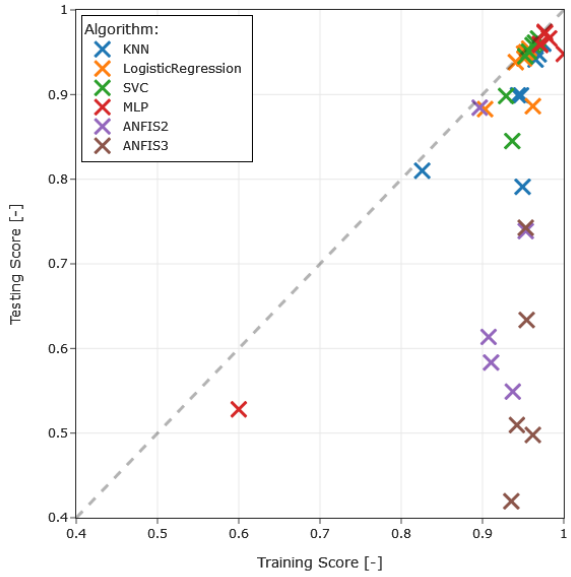

Fig. 7. Scatter plot of testing score against training score by algorithm.

TABLE IV. EXPERIMENT RESULTS TABLE

Algorithm	Training Sample	Training Accuracy	Testing Accuracy	F1 Score	Overfit Score
ANFIS2	156	0.948	0.285	0.203	0.664
ANFIS2	1561	0.953	0.739	0.727	0.215
ANFIS2	15610	0.908	0.614	0.575	0.294
ANFIS2	78	0.962	0.346	0.284	0.616
ANFIS2	7805	0.911	0.583	0.597	0.327
ANFIS2	78052	0.897	0.885	0.888	0.012
ANFIS2	781	0.937	0.549	0.519	0.388
ANFIS3	156	0.935	0.419	0.429	0.516
ANFIS3	1561	0.953	0.743	0.742	0.211
ANFIS3	78	0.962	0.498	0.516	0.464
ANFIS3	7805	0.954	0.634	0.605	0.321
ANFIS3	781	0.942	0.510	0.499	0.433
KNN	156	0.826	0.810	0.798	0.016
KNN	1561	0.944	0.899	0.899	0.044
KNN	15610	0.969	0.948	0.948	0.022
KNN	78	0.949	0.791	0.774	0.158
KNN	7805	0.965	0.941	0.941	0.024
KNN	78052	0.975	0.960	0.960	0.015
KNN	781	0.948	0.899	0.899	0.048
Logistic Regression	156	0.903	0.883	0.883	0.020
Logistic Regression	1561	0.941	0.938	0.938	0.003
Logistic Regression	15610	0.954	0.950	0.950	0.004
Logistic Regression	78	0.962	0.886	0.886	0.076
Logistic	7805	0.951	0.948	0.948	0.003

Algorithm	Training Sample	Training Accuracy	Testing Accuracy	F1 Score	Overfit Score
Regression					
Logistic Regression	78052	0.957	0.954	0.954	0.003
Logistic Regression	781	0.951	0.943	0.942	0.009
MLP	156	0.600	0.528	0.375	0.072
MLP	1561	0.971	0.959	0.958	0.013
MLP	15610	0.977	0.973	0.973	0.004
MLP	78	1.000	0.948	0.948	0.052
MLP	7805	0.982	0.966	0.966	0.016
MLP	78052	0.976	0.974	0.974	0.002
MLP	781	0.971	0.960	0.960	0.011
SVC	156	0.929	0.898	0.898	0.031
SVC	1561	0.957	0.951	0.951	0.006
SVC	15610	0.965	0.961	0.961	0.003
SVC	78	0.937	0.845	0.841	0.092
SVC	7805	0.962	0.958	0.958	0.003
SVC	78052	0.968	0.966	0.966	0.002
SVC	781	0.953	0.945	0.945	0.007

VII. CONCLUSIONS & FURTHER WORK

It is concluded that ANFIS did not offer any benefit over conventional machine learning algorithms in this application, and that MLP was the most consistent across all training sample sizes. It is also concluded that Logistic Regression performed very well against less biased algorithms.

The greatest limitation in the set of experiments was the handling of training validation to minimize the risk of over-fitting the ANFIS models. The conventional algorithms benefited from ten-fold cross validation whereas the ANFIS algorithm was only provided a hold-out sample with an epoch number estimate.

The computation times on solving the ANFIS models caused some real challenges, parallel computing would need to be employed to keep training times reasonable. Future work should either look to find a way to enable parallel computation.

The smaller training samples may also have suffered from the random split, which had a fixed seed, rather than cross-validation with multiple seeds – though stratification does go somewhat towards managing this issue by maintaining the same class imbalance in each sample.

REFERENCES

Bai, Y., Liu, J., Wang, S., & Yang, F. (2018). Machine Learning Applied to Star-Galaxy-QSO Classification and Stellar Effective Temperature Regression. *The Astronomical journal*, *157*(1), 9. https://doi.org/10.3847/1538-3881/aaf009

Beitia-Antero, L., Yáñez, J., & de Castro, A. I. G. (2018). On the use of logistic regression for stellar classification: An application to colour-colour diagrams. *Experimental astronomy*, *45*(3), 379-395. https://doi.org/10.1007/s10686-018-9591-4

Fedesoriano. (2022). *Stellar Classification Dataset - SDSS17*. https://www.kaggle.com/fedesoriano/stellar-classification-dataset-sdss17.

Garcia-Dias, R., Prieto, C. A., Almeida, J. S., & Ordovás-Pascual, I. (2018). Machine learning in APOGEE. *A&A*, *612*, A98. https://doi.org/10.1051/0004-6361/201732134

Kovács, A., & Szapudi, I. (2015). Star-galaxy separation strategies for WISE-2MASS all-sky infrared galaxy catalogues. *Monthly Notices of the Royal Astronomical Society*, *448*, 1305-1313. https://doi.org/10.1093/mnras/stv063

Krakowski, T., Małek, K., Bilicki, M., Pollo, A., Kurcz, A., & Krupa, M. (2016). Machine-learning identification of galaxies in the WISE × SuperCOSMOS all-sky catalogue. *Astronomy and Astrophysics*, *596*, A39. https://doi.org/10.1051/0004-6361/201629165

Mohana Sundari, L., & Sivakumar, P. (2021). Detection and Segmentation of Cracks in Weld Images Using ANFIS Classification Method. *Russian journal of nondestructive testing*, *57*(1), 72-82. https://doi.org/10.1134/S1061830921300033

Nagarathinam, E., & Ponnuchamy, T. (2019). Image registration ‐ based brain tumor detection and segmentation using ANFIS classification approach. *International journal of imaging systems and technology*, *29*(4), 510-517. https://doi.org/10.1002/ima.22329

Talpur, N., Salleh, M. N. M., & Hussain, K. (2017). An investigation of membership functions on performance of ANFIS for solving classification problems. *IOP conference series. Materials Science and Engineering*, *226*(1), 12103. https://doi.org/10.1088/1757-899X/226/1/012103

You, H., Ma, Z., Tang, Y., Wang, Y., Yan, J., Ni, M., Cen, K., & Huang, Q. (2017). Comparison of ANN (MLP), ANFIS, SVM, and RF models for the online classification of heating value of burning municipal solid waste in circulating fluidized bed incinerators. *Waste management (Elmsford)*, *68*, 186-197. https://doi.org/10.1016/j.wasman.2017.03.044

Zhang, J., He, Z. Y., Lin, S., Zhang, Y. B., & Qian, Q. Q. (2013). An ANFIS-based fault classification approach in power distribution system. *International journal of electrical power & energy systems*, *49*, 243-252. https://doi.org/10.1016/j.ijepes.2012.12.005

Zhao, Z., Wei, J., & Jiang, B. (2022). Automated Stellar Spectra Classification with Ensemble Convolutional Neural Network. *Advances in astronomy*, *2022*, 1-7. https://doi.org/10.1155/2022/4489359