Exoplanet Detection: Efficient Search using Data Mining Techniques

Raghav Mehta
School of Computing (MSc in Data Analytics)
National College of Ireland
Dublin, Ireland
x17155151@student.ncirl.ie

Abstract— **Exoplanets are defined as planets orbiting a star outside our solar system. There have been considerable improvements over the past decades in the detection of exoplanets. NASA launched the Kepler satellite in 2009 with the mission to hunt exoplanets. The data gathered by the satellite have been made publicly available by NASA in an attempt to increase development in the field of astroinformatics, a cross-disciplinary field consisting of astronomy, data science and informatics. This has led to increased contribution in detecting exoplanets among non-exoplanets. In this paper, various techniques to detect exoplanets using the intensity of light captured by Kepler space telescope have been explored. The dimming in flux (light intensity) of the stars indicates an orbiting body around that star. Preliminary analysis is performed and the data is prepared to be fed in to the data mining models. Multiple models have been used to efficiently classify the exoplanets. The first model, H2O's Deep Learning, is based on a multi-layer feedforward artificial neural network. Then Gradient Boosting technique is performed using XGBoost algorithm, an ensemble technique that works on the concept of Decision Tree and Bootstrap Aggregation. Finally, a Convolutional Neural Network, a Black Box method, is deployed to classify the exoplanets. This paper will provide the results, model performance and efficiency of the various algorithms used to detect such exoplanets.**

Keywords—Machine Learning, Gradient Boosting, Principal Component Analysis, Random Forest, Exoplanets, Neural Networks, XGBoost Algorithm, H20 Deep Learning, Convolutional Neural Network, Feature Selection

I. Introduction

The first evidence of finding exoplanets dates back to 1917, followed by scientific detection in 1988. But 1992 gave a new light with the first affirmative detection of an exoplanet. With the development of ground based and space borne observatories, thousands of exoplanets have been discovered. Today there are about 3797 confirmed planets in 2841 star systems. A planet under the gravitational influence of a star orbits around it. The planets do not emit any light of their own and hence can only be detected by capturing the flux of the star. When the Kepler satellite, star and the astronomical body are closely aligned, the astronomical body between the satellite and the star, there is a considerable decrease in the flux captured. This decrease in light intensity is evidence of a body orbiting the star, together called a 'candidate system'. By analyzing the decline in brightness of the star over a period of time we can confirm the presence of a exoplanet in the candidate system.

The Kepler Mission, launched in 2009, has monitored the light intensity of more than 150,000 stars. The aim of the Kepler mission is to find planet similar to Earth in the habitable zone of star like the Earth-Sun system. The telescope deployed upon the Kepler satellite is of 0.95m diameter and has a resolution of 2200 * 1024 pixels. Total resolution of the telescope is 94.6 megapixels, giving it a 115 sq. degrees view of space. The data gathered by the satellite have been made publicly available by NASA in an attempt to increase development in the field of astroinformatics, a cross-disciplinary field consisting of astronomy, data science and informatics. This has led to increased contribution in detecting exoplanets.

The project is an intersection of cosmology and machine learning, the primary motivation behind choosing this project. There are also challenging tasks arising due to the large dimension of data and class imbalance. We also get the chance to explore multiple models as well as use additional techniques of cross validation and hyperparameter optimization. Hence there is substantial learning in this project, empowering us in developing models on real world problems later on. The objective of the project is to apply deep learning techniques in finding the existence of exoplanet. The dimming of flux may not necessarily occur due to the presence of a planet, and may be caused by some other space object. Our task is to differentiate amongst candidate systems and space objects, and confirm the presence of exoplanet in the candidate system. This is achieved by reducing the dimension of the data, followed by generating multiple unique models, parameter tuning of models and finally validating the model.

The paper is divided in 5 sections. Section 2 consists of related works done in this field. In Section 3 is an elaboration on the data mining methodology used in this project along with all the preprocessing and data preparation prior to building our model. In Section 4, the architecture of the model is discussed and the results generated are interpreted in depth. The paper concludes with the final section summarizing the findings and future scope in this research area.

II. Related Works

Learning about candidate planet started from the Gaussian method. It is a statistical approach using probability of detection and false alarm. In this method a moment generating function is used which is a highly accurate approximation formula, called saddle point approximation [1]. The same year Kepler mission was launched to increase contribution in finding Earth-like planets in habitable zone of Sun-like stars. This project was planned to work for 3.5 years to gather data and establishing other scientific approaches in finding exoplanets with the help of technologies and machinery present during that time such as a huge telescope and scientific research [2]. The data captured by the telescope gave huge amounts of insight to humans. This was achieved by preprocessing the data using computers, either mainframe or desktop. So it became the second most important tool for deriving results out of data [3].

Amongst the techniques used in search of candidate systems, extra solar planet radical velocity is the most widely used. In this approach the wandering of a star and planet is captured using Doppler shift of light from sun-like star [4].

Principal component analysis and image processing, with 3 or more principal components, has also been widely used besides the radial velocity approach. This is done since flux of exoplanets are recorded in more than 3000 positions [5].

Unsupervised Bayesian approach detects modes of oscillation and mode frequencies. Using frequencies parameters of sun like stars are obtained like their ratios and covariance matrix [6].

Even though number of advanced researches and models have been constructed using Doppler effect, radial velocity and spectrographic & transit detection, we are still in the initial stage of finding such exoplanets [7].

Amongst the above techniques the most commonly used is the transit detection method. Detection of exoplanets transiting in front of the telescope are calculated in the data science field [8].

ASTROMLSKIT, a toolkit embedded with machine learning algorithms, was developed in year 2015. Accuracy given by Naïve Bayes and Decision Tree (98.86%) , KNN (96.59%) and Random Forest (97.72%) are quite remarkable. This is only made possible after massive research, gathering large amounts of data and the primary challenge of clearing out noise from the data [9].

Robust PCA (Principle Component Analysis and PCA algorithms made it feasible to reduce the dimension of sequence received from images. PCA uses the method of Singular Value Decomposition to reduce the dimensions of data. Low-rank plus sparse (LRPS) decomposition is also another method for the same, which is found to have higher performance compared to PCA [10].

When exoplanets hunting started in late 1992 there were limited technologies which can learn from data and give some insight but today with new era and advent of artificial intelligence and bigdata there are many changes seen in detection of candidate planet [11].

Noise within data is the main problem while detecting exoplanets. Bootstrap approach is used commonly to avoid false selection while the search for exoplanets. Exoplanets are highly irregular, with unknown statistics and coloured noise, hence this method has gained popularity[12].

Convolutional Neural Network, a deep learning algorithm, also came into picture and opened gates for detecting exoplanets. When trained with exoplanets dataset to identify earth like planets, the CNN algorithm was able to positively identify two exoplanets with an accuracy of 98.8% [13].

Digital cameras and high definitions lens used in telescopes increased the data volume with high pace and made Big data available for analysis. Today machine learning and huge volumes of data available attracts more and more data scientists and makes astronomy area open for research [14].

Manual interpretation of bigdata is labour intensive and not easy to perform, specifically with transit signals which are different for exoplanets. Machine learning made it feasible to learn from such huge volume of cosmological datasets [15].

Radio signals coming from exoplanets is also a new area of research for exoplanets. Though there has been no success in detecting an exoplanet using radio signal algorithms, this is due to the noise and distance of exoplanets. So the most commonly used techniques are still radial velocity, microlensing and transit system [16].

Finding Earth like planet in far solar-like stars and their habitable zone, has also opened the field of finding Mars like planet by taking candidate exoplanets into consideration [17].

Class imbalance is the main issue when it comes to exoplanets datasets. It effects the possibility of exoplanet detection using machine learning. So to detect exoplanets accurately we need to create a derived field out of given data and oversample over training data using random forest, decision tree, xgboost. These models drastically affect the accuracy of exoplanet predicting algorithms [18].

Effective machine learning algorithm or techniques are limited today. But neural networks a simple deep learning algorithm is efficient and one of the most widely used method in predictive analysis in such scenarios [19].

With considerable improvement in search of candidate exoplanet, researchers are interested in finding the most optimized algorithm to correctly detect extra-solar planet using Kepler mission data and flux of dimming light when planet orbits around habitable zone of sun-like star [20].

Moving forward with vast work already done in area of Extra-solar Planet research (exoplanets), in this document we will apply PCA on the data, construct the algorithms and optimize them.

III. Data Mining Methodology

For our research, we have used the Knowledge Discovery in Databases (KDD) methodology. The term Data Mining and KDD are often used interchangeably. This is incorrect as Data Mining is just one part of the KDD process. KDD refers to a framework which includes multiple steps to derive knowledge from data [21]. Our focus is on finding underlying patterns from the data, which includes statistical analysis and machine learning techniques. Hence KDD is the preferred approach for this research.

KDD is a five step process that includes **selection** of the target data, **preprocessing** to clean and subsample the data, **transformations** on data as required, **data mining** to find patterns based on a specific technique e.g., classification, regression, clustering, etc. and finally **interpretation** of the results for evaluating the degree of correctness of the model to the applied problem [22]. We discuss the first 4 steps in this section followed by the evaluation of models and interpretation in the next section.

A. *Selection*

The dataset for our research is open-sourced, made publicly available by NASA. The data has a total of 5657 records and 3197 flux values which are numeric in nature [23]. Our dependent variable is a categorical variable – whether a star has a confirmed exoplanet or not. We have retained all the features and instances for our model.

B. *Preprocessing*

The curvature of the Kepler satellite's telescope adds noise to the data, which is then beamed down to Earth. NASA applies de-noising algorithms to remove these irregular fluctuations from the data before making it public. Our data does not contain missing values and duplicate rows either.

C. *Transformation*

Since we have 3197 independent continuous features, it is infeasible to draw a correlation matrix and analyze the flux correlation. Therefore we have drawn two scatter plot between two randomly selected flux values.

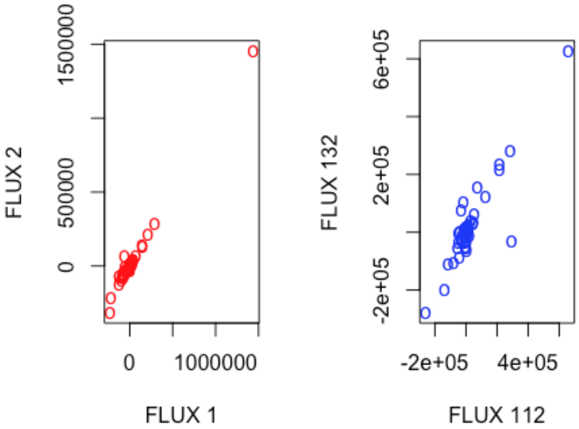


Fig. 1. Scatter Plot Between FLUX variables

We first perform feature selection to extract the relevant features from our dataset. We apply the Random Forest ensemble learning method on our classification problem. We measure the importance of the variables in this setting using the mean decrease of the Gini index [24].

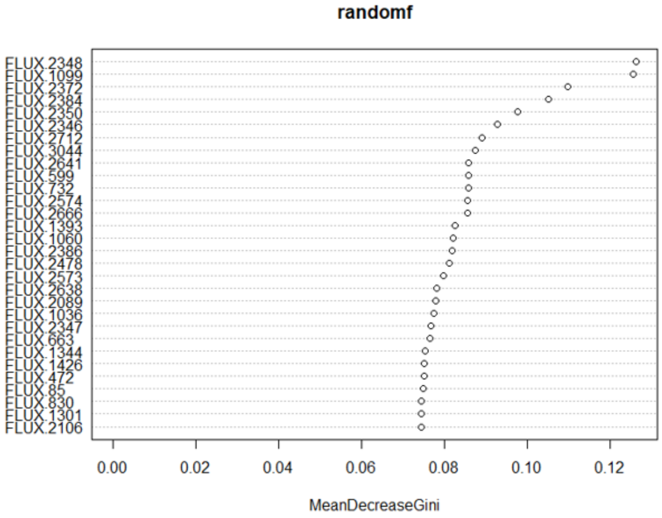
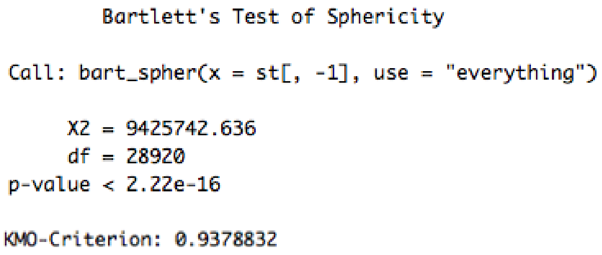


Fig. 2. Rank of important features

The greater the decrease in accuracy due to exclusion of the variable, the more important that variable is considered. We retain only those features with the criterion value greater than 0.05. This reduces our feature count from 3197 to 241.

Since our features are linearly correlated, it justifies the need for dimensionality reduction. We perform the Principal Component Analysis technique for factor reduction. Preliminary analyses for PCA are conducted to confirm the factorability of the data. This is done using Bartlett's Test of Sphericity and the Kaiser-Meyer-Olkin measure. As seen in Fig. 3., Bartlett's test is significant ($p < .05$) and KMO index is greater than 0.6. Hence factor analysis using PCA is appropriate and suggested good [25]. PCA is explained through Singular Value Decomposition of matrix. It leads to higher numerical accuracy which is preferred. Using the Kaiser criteria of choosing eigenvalues greater than 1, we are left with 15 components [26].



```
         Bartlett's Test of Sphericity

Call: bart_spher(x = st[, -1], use = "everything")

     X2 = 9425742.636
     df = 28920
p-value < 2.22e-16

KMO-Criterion: 0.9378832
```

Fig. 3. Statistical Measures for Factor Analysis to be appropriate

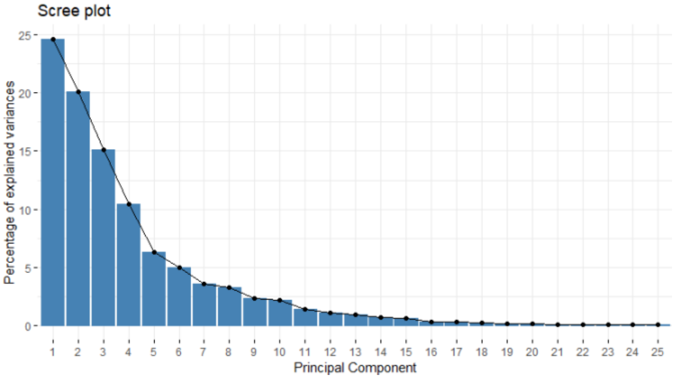


Fig. 4. PCA Screeplot

The data is scaled and centred while running PCA algorithm. This standardization converts mean 0 and standard deviation 1 for each feature. Centering data reduces the square mean error of approximating the feature data whereas scaling is done to ensure that the data is measured on similar scale for all features [27][28].

Fig. 5. shows the kernel density estimates for two different principal components and each of the classes of the target variable. It can be seen that there is no fluctuation in the distribution of the PCA when class is 0 ie. non-exoplanet. However there are local maxima's in the density distribution in cases where there are confirmed exoplanets. This verifies the results of PCA are suitable and as expected.

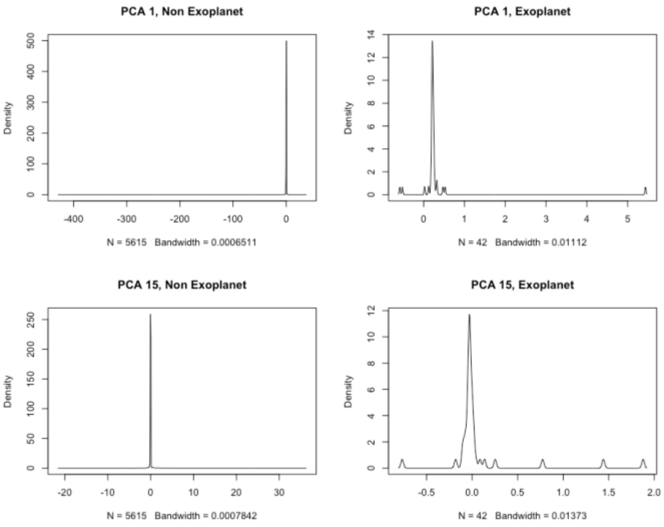


Fig. 5. Density Plots

After completing factor analysis, we partition our data in to train and test sets with a ratio of 8:2. We have chosen to partition our data before dealing with the imbalance in target variable class'. This is done to ensure that none of the data in the test set is randomly generated and consists only of the actual data, which is further used to compute the final model performance and accuracy.

The class balance of our target variable is 99.26%. Hence it is imperative to balance the class before modelling the transformed data. We have chosen the smoothed bootstrap technique via ROSE package for this task. While SMOTE method tends to overfit the data, ROSE generates new artificial examples. The classifier derived after sampling using ROSE, while tends to overestimate the accuracy, has a low mean square error [29]. Class balancing is performed only on the training data and not on the testing data. This is done to ensure that the classifier created does not overfit the data and we get a honest estimation of our model.

Using ROSE, we have oversampled the minority class and undersampled the majority class, but kept the total number of records at 5657 only. The class balance is now 2:1 by changing the minority class from 0.74% to 33%.

The final step in transformation is to convert the categorical dependent variable to binary classification. While our target variable is dichotomous in nature with factor levels 1 and 2, it is necessary for us to convert it to binary levels of 0 and 1. This is required as certain machine learning models only accept binary inputs.

D. *Data Mining*

We have generated three different models for our problem. These are: H2O Deep Learning based on multi-layer feed forward artificial neural network, Convolutional Neural Network and Gradient Boosting using the XGBoost algorithm.

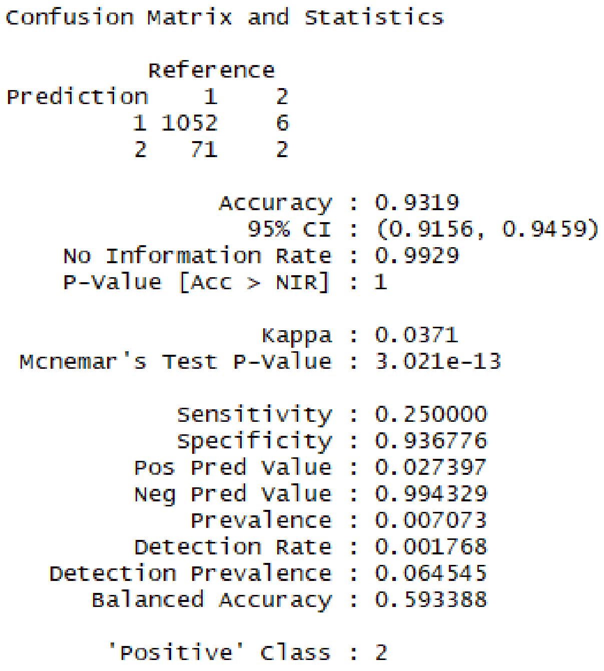
IV. Model Construction and Evaluation

A. *H2O Deep Learning*

This model works on a multi-layer feedforward artificial neural network. Simply using Artificial Neural Network is not feasible because of the extremely high computation time and the complexity of the model due to the higher number of neurons and layers. H2O helps to scale our network and reduce computation time. It also allows us to perform hyperparameter optimization. Hence we can find the required combination of hyper parameters which can increase efficiency of our performance matrix such as accuracy.

We start by connecting the R studio to H2O cluster at 2.30 GB cluster memory. We then create a deep learning ANN classifier at 100 epochs taken randomly. This is done with the hope that the model converges. For the model, we have dedicated 3 hidden layers with 32 neurons each. The stopping metric used is "*misclassification*" as we are dealing with a classification problem. For our untuned model, the activation function used is the default one, that is the Rectifier.

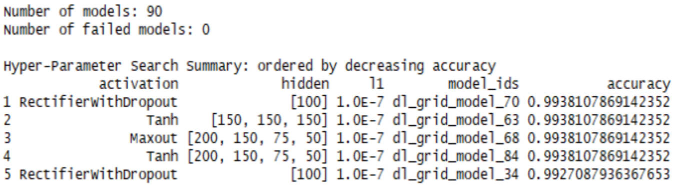
The model accuracy is 93.19% with 71 non exoplanets and 6 exoplanets being wrongly classified. The F-measure for the model is 0.964.



```
Confusion Matrix and Statistics

          Reference
Prediction    1    2
         1 1052    6
         2   71    2

               Accuracy : 0.9319
                 95% CI : (0.9156, 0.9459)
    No Information Rate : 0.9929
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0371
 Mcnemar's Test P-Value : 3.021e-13

            Sensitivity : 0.250000
            Specificity : 0.936776
         Pos Pred Value : 0.027397
         Neg Pred Value : 0.994329
             Prevalence : 0.007073
         Detection Rate : 0.001768
   Detection Prevalence : 0.064545
      Balanced Accuracy : 0.593388

       'Positive' Class : 2
```

Fig. 6. H2O Neural Network Confusion Matrix

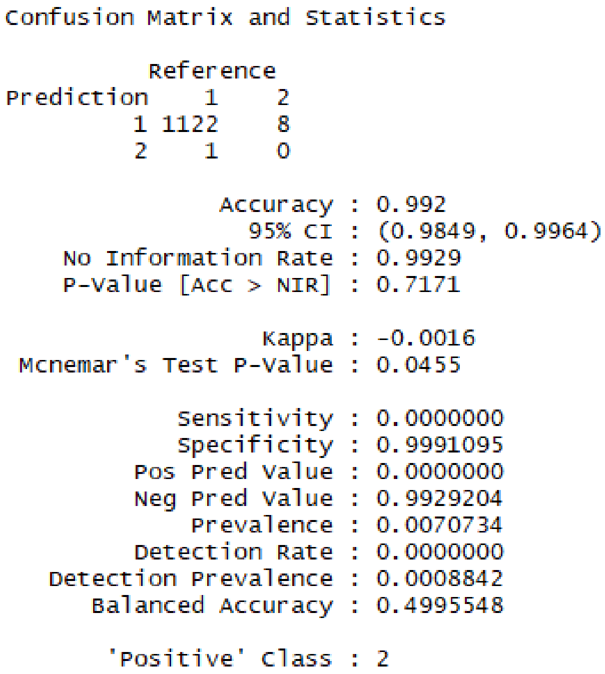
We now look at hyperparameter optimization. We need to perform hyperparameter tuning so that we can improve the model and find which all hyperparameters serve best for our model. This is achieved through Grid Search optimization, where we sail through different ranges of parameters. Tuning these parameters will help us to find the best fit model based on our performance metric. The hyperparameters which we have tuned are the number of hidden layers, size of layers (neurons) and the activation function. The regularization parameter is also used which applies a penalty when the model is overfitted. A total of 90 models were generated through this method.



```
Number of models: 90
Number of failed models: 0

Hyper-Parameter Search Summary: ordered by decreasing accuracy
           activation              hidden      l1         model_ids           accuracy
1 RectifierWithDropout              [100] 1.0E-7 dl_grid_model_70 0.9938107869142352
2                Tanh     [150, 150, 150] 1.0E-7 dl_grid_model_63 0.9938107869142352
3              Maxout [200, 150, 75, 50] 1.0E-7 dl_grid_model_68 0.9938107869142352
4                Tanh [200, 150, 75, 50] 1.0E-7 dl_grid_model_84 0.9938107869142352
5 RectifierWithDropout              [100] 1.0E-7 dl_grid_model_34 0.9927087936367653
```

Fig. 7. Hyperparameter Model (Top 5) Summary

The best fit model by tuning the hyperparameters led to an accuracy of 99.2% with only one non-exoplanet being misclassified. However all the 8 exoplanets in the testing data were wrongly classified. The activation function used is RectifierwithDropout, for which the model returned the highest accuracy. The F-measure for this model is 0.996.



```
Confusion Matrix and Statistics

          Reference
Prediction    1    2
         1 1122    8
         2    1    0

               Accuracy : 0.992
                 95% CI : (0.9849, 0.9964)
    No Information Rate : 0.9929
    P-Value [Acc > NIR] : 0.7171

                  Kappa : -0.0016
 Mcnemar's Test P-Value : 0.0455

            Sensitivity : 0.0000000
            Specificity : 0.9991095
         Pos Pred Value : 0.0000000
         Neg Pred Value : 0.9929204
             Prevalence : 0.0070734
         Detection Rate : 0.0000000
   Detection Prevalence : 0.0008842
      Balanced Accuracy : 0.4995548

       'Positive' Class : 2
```

Fig. 8. Confusion Matrix of best fit tuned model

B. *Convolutional Neural Network*

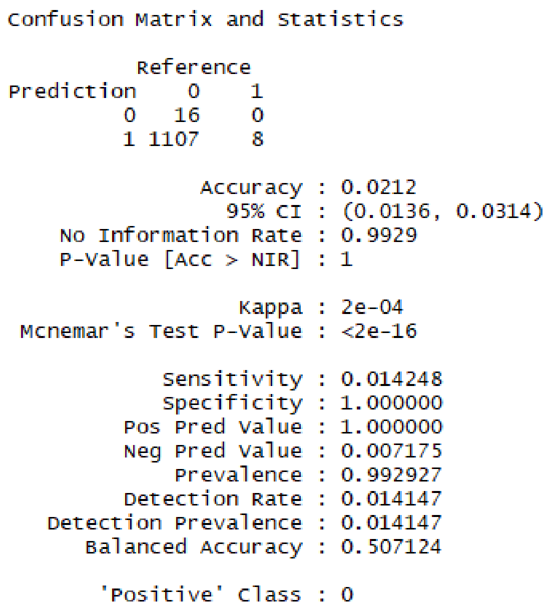
Keras comes into picture when we need to implement Deep Learning algorithms. It is driven by Python library with TensorFlow at its backend. To run Keras in R, an instance of Conda needs to be created which installs and runs all the Python libraries. CNN is majorly used for image and object classification. It consists of a convolution layer, pooling layer, fully connected layer and a normalization layer. As we are not dealing with image data, so we transform a 2D convolutional layer to a 1D convolutional layer.

Firstly we reshaped the Principal Components in to an set arrangement of array and performed one-hot encoding of the labels. Then we used a 1D convolutional layer consisting of 10 filters and Relu as the activation function with a kernel size of 10. Relu is used as an activation function because of it's linear nature and the flexibility it provides by choosing to run or not run it.

Finally we use softmax activation function in the final layer so that the output of the neural network is converted to a probability distribution. We have used a batch size of 30 while fitting the model so that it does not occupy a large memory space.

The training data was divided in to 8:2, with 20% validation data. The model resulted in extremely high accuracy and kappa statistic when validating the model, with minimum loss.

When testing the final model with authentic highly imbalanced test data, the performance of the model was impacted drastically. The accuracy of the model was 2% with F-measure of 0.028.



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0   16    0
         1 1107    8

               Accuracy : 0.0212
                 95% CI : (0.0136, 0.0314)
    No Information Rate : 0.9929
    P-Value [Acc > NIR] : 1

                  Kappa : 2e-04
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.014248
            Specificity : 1.000000
         Pos Pred Value : 1.000000
         Neg Pred Value : 0.007175
             Prevalence : 0.992927
         Detection Rate : 0.014147
   Detection Prevalence : 0.014147
      Balanced Accuracy : 0.507124

       'Positive' Class : 0
```

Fig. 9. CNN confusion matrix

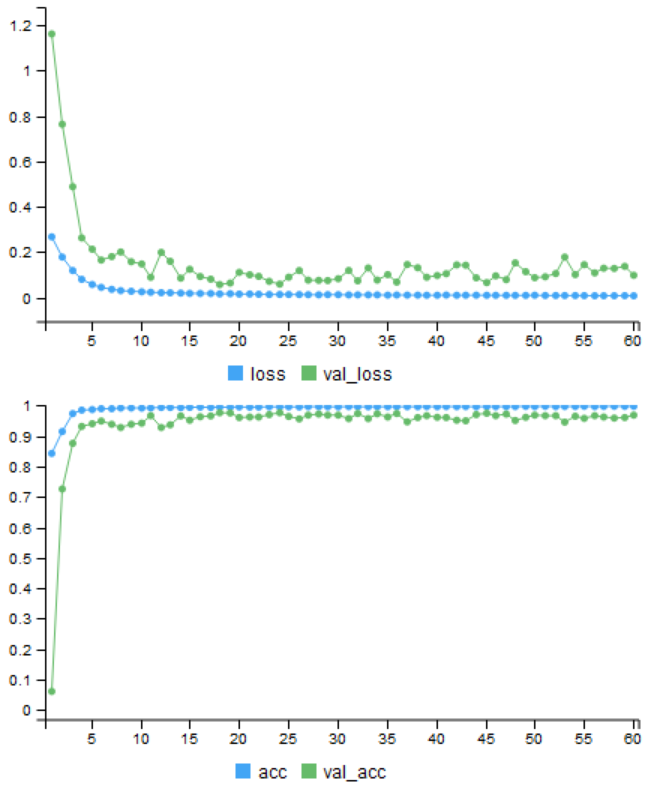


Fig. 10. Accuracy and Loss graphs for Training and Validation Sets

C. Gradient Boosting using XGBoost Algorithm

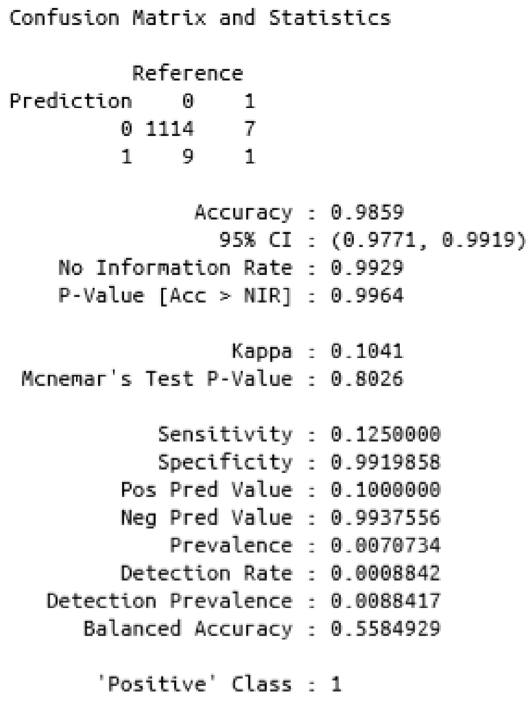
XGBoost is considered one of the most efficient, feasible and accurate algorithm used throughout. It is both a linear model and a tree learning algorithm, and due to the parallel computation capabilities, it runs extremely fast even on a single machine. XGBoost provides additional features that help us in performing cross validation and identifying important features.

XGBoost works only with numerical data and hence comes as our first choice for its selection due the fact our dataset is complexly numerical. XGBoost converts categorical value into numeric vector using 1 hot encoding method and thus can be adapted easily. XGBoost is available in different packages and the one chosen for this project is from CARET package. In Table 1, the parameters used to tune the XGBoost model using grid search optimization are listed.

TABLE I. Tuning Parameters

Parameter	**Argument values Passed**
Depth	(50)
Rounds	(50,100,500)
Learning rate/eta	(0.01,0.3)
Colsample_bytree	(0.5-0.9)
Gamma	(0,1)
CV	5

The model is trained with best hyperparameter tuning settings resulted from the control and grid. These are Fitting nrounds = 500, max_depth = 10, eta = 0.01, gamma = 0, colsample_bytree = 0.7, min_ child_weight = 0.85, subsample = 1 on full training set.



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1114    7
         1    9    1

               Accuracy : 0.9859
                 95% CI : (0.9771, 0.9919)
    No Information Rate : 0.9929
    P-Value [Acc > NIR] : 0.9964

                  Kappa : 0.1041
 Mcnemar's Test P-Value : 0.8026

            Sensitivity : 0.1250000
            Specificity : 0.9919858
         Pos Pred Value : 0.1000000
         Neg Pred Value : 0.9937556
             Prevalence : 0.0070734
         Detection Rate : 0.0008842
   Detection Prevalence : 0.0088417
      Balanced Accuracy : 0.5584929

       'Positive' Class : 1
```

Fig. 11. XGBoost Confusion Matrix

This model is further applied on the training dataset and the accuracy achieved in around 98% however due to the fact the dataset is highly imbalanced we can rule out the accuracy achieved by the model. As per the results generated using the confusion matrix, the Kappa score is significantly low at 10.41% as well as sensitivity at just 12.5%. The specificity for the model is really high at 99%. The model was however able to predict 1 exoplanet out of 8 exoplanets available in testing dataset and correctly classified 1114 non-exoplanets out of 1125 exoplanets.

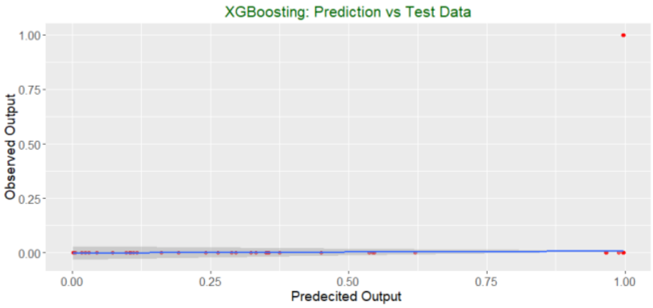


Fig. 12. Predicted vs Test Data Classification

V. Conclusion And Future Scope

Our Deep Learning H2O neural network model has a performance accuracy of 93.19% with a very low Kappa statistic. The model was only able to classify 25% (sensitivity) of the exoplanets. It was highly successful in classifying the non-exoplanets leading to a modestly high specificity. However after performing hyperparameter optimization, we were able to capture a model with accuracy of 99.2%. This however was at the cost of a lower Kappa statistic. The sensitivity of the model came down to 0 ie. it was unable to classify even a single exoplanet. However the prediction of non-exoplanets improved with a specificity of 99.9%. Only 1 out of 1123 non-exoplanets were misclassified.

The primary reason behind such model statistics arose due to the nature of the data. Since initially we started with a class imbalance of 99.3% to 0.7%, and then synthetically sampled the minority class of exoplanets, the model tends to become biased towards such generated data of the minority class. The tuning was done on monitoring factor of accuracy, which led to a selection of model with least misclassification, whilst sacrificing the prediction of presence of exoplanet in the data.

The CNN model performed poorly in terms of accuracy (2%), even though it was able to classify all the exoplanets. On the contrary, the model was unable to predict the non-exoplanets with a low sensitivity of 1.4%.

While training the model, the validation dataset of 10% was correctly classified. This happened as most of the minority class data in the training and validation set were randomly sampled together. Hence even though the validation set lead to a high accuracy for our model, when the model was used on the original Kepler's test data it was highly inefficient to classify the data.

Gradient Boosting model gave a relatively high accuracy of 98% with a low Kappa statistic of 10.41%. The best measure to evaluate the model is the sensitivity and specificity. The model only misclassified 9 non-exoplanets with a specificity of 99.2%. The model also classified 1 out of 8 exoplanets in the testing data.

Due to the high class imbalance and random oversampling, none of the models provide conclusive evidence in the existence of exoplanets specifically. Gradient Boosting model performed the best out of all the 3 machine learning models, even after tuning the neural networks.

There are multiple improvements that can be made to these models. Ideally the data should be less imbalanced to begin with. We can apply other feature selection methods such as Boruta algorithm, vida algorithm or perform independent component analysis. These are extensive methods of performing feature engineering using random forest as well.

We also face limitation in terms of computation power available. GPU will help in parallel computing operations, hence resulting in significant performance gains.

Another drawback of low computational power is that hyperparameter optimization cannot be run in the ideal fashion. It does not follow a set algorithm, so ideally its encouraged to use all available hyperparameters and choose the model which monitors not just high accuracy, but also the sensitivity and specificity. Improvement can also be increased by having more data points which correspond to existence of exoplanets.

As CNN was highly specific in classifying the exoplanets, we propose an ensemble model which uses the prediction of H2O Deep Learning Neural Network and Convolutional Neural Network to generate a more efficient classifier.

It is still hard to classify whether a star harbours an exoplanet. There are other models which might improve the performance in terms of detection of exoplanets. Models such as ARIMA, Hidden Markov or Neuro Fuzzy Hybrid Systems can be experimented with in the future.

References

[1] I. Smith, A. Ferrari, and M. Carbillet, "Detection and performance analysis for a moving point source in speckle noise, application to exoplanet detection by direct imaging," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, pp. 3661–3664, 2008.

[2] W. Borucki *et al.*, "KEPLER: Search for earth-size planets in the habitable zone," *Proc. Int. Astron. Union*, vol. 4, no. S253, pp. 289–299, 2008.

[3] J. Kanipe, "Modeling the astronomical," *Commun. ACM*, vol. 53, no. 5, p. 13, 2010.

[4] P. O. Box, "A COMBINED LINEAR PROGRAMMING-MAXIMUM LIKELIHOOD APPROACH TO RADIAL VELOCITY DATA ANALYSIS FOR EXTRASOLAR PLANET DETECTION, Division of Systems and Control , Department of Information Technology," no. 4, pp. 4352–4355, 2011.

[5] A. Amara and S. P. Quanz, "PYNPOINT: An image processing package for finding exoplanets," *Mon. Not. R. Astron. Soc.*, vol. 427, no. 2, pp. 948–955, 2012.

[6] G. R. Davies *et al.*, "Oscillation frequencies for 35 Kepler solar-type planet-hosting stars using Bayesian techniques and machine learning," *Mon. Not. R. Astron. Soc.*, vol. 456, no. 2, pp. 2183–2195, 2015.

[7] D. A. Fischer *et al.*, "Exoplanet Detection Techniques," 2015.

[8] K. Heng and J. Winn, "The Next Great Exoplanet Hunt," vol. 103, 2015.

[9] Snehanshu Saha, Surbhi Agrawal, Manikandan. R, Kakoli Bora, Swati Routh, and Anand Narasimhamurthy, "ASTROMLSKIT:

A New Statistical Machine Learning Toolkit: A Platform for Data Analytics in Astronomy," *eprint arXiv:1504.07865*, 2015.

[10] C. A. Gomez Gonzalez, O. Absil, P.-A. Absil, M. Van Droogenbroeck, D. Mawet, and J. Surdej, "Low-rank plus sparse decomposition for exoplanet detection in direct-imaging ADI sequences," *Astron. Astrophys.*, vol. 589, p. A54, 2016.

[11] A. Ai, "Machines learning.," *Economist*, pp. 53–54, 2016.

[12] S. Sulis, D. Mary, and L. Bigot, "A bootstrap method for sinusoid detection in colored noise and uneven sampling. Application to exoplanet detection," *25th Eur. Signal Process. Conf. EUSIPCO 2017*, vol. 2017–Janua, pp. 1095–1099, 2017.

[13] C. J. Shallue and A. Vanderburg, "Identifying Exoplanets with Deep Learning: A Five Planet Resonant Chain around Kepler-80 and an Eighth Planet around Kepler-90," *Astron. J.*, vol. 155, no. 2, p. 94, 2017.

[14] J. Kremer, K. Stensbo-Smidt, F. Gieseke, K. S. Pedersen, and C. Igel, "Big Universe, Big Data: Machine Learning and Image Analysis for Astronomy," pp. 1–15, 2017.

[15] K. A. Pearson, L. Palafox, and C. A. Griffith, "Searching for exoplanets using artificial intelligence," *Mon. Not. R. Astron. Soc.*, vol. 474, no. 1, pp. 478–491, 2018.

[16] M. J. Bentum, "Algorithms for Direct Radio Detections of Exoplanets in the Neighbourhood of Radiating Host Stars," *IEEE Aerosp. Conf.*, pp. 1–7, 2018.

[17] M. Kashyap Jagadeesh, S. B. Gudennavar, U. Doshi, and M. Safonova, "Indexing of exoplanets in search for potential habitability: application to Mars-like worlds," *Astrophys. Space Sci.*, vol. 362, no. 8, 2017.

[18] S. Basak *et al.*, "Habitability Classification of Exoplanets: A Machine Learning Insight," 2018.

[19] S. Butte, A. R. Prashanth, and S. Patil, "08360836."

[20] K. Mohamed, A. Ibrahim, and N. Abd-Allah, "An optimized search for exoplanets with Kepler data," *J. Phys. Conf. Ser.*, vol. 869, no. 1, 2017.

[21] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "Knowledge Discovery and Data Mining: Towards a Unifying Framework.," *Int Conf Knowl. Discov. Data Min.*, pp. 82–88, 1996.

[22] A. Azevedo and M. F. Santos, "KDD, SEMMA and CRISP-DM: a parallel overview," *IADIS Eur. Conf. Data Min.*, no. January, pp. 182–185, 2008.

[23] Kaggle, "Kepler Labelled Time Series Data," 2017. .

[24] F. Degenhardt, S. Seifert, and S. Szymczak, "Evaluation of variable selection methods for random forests and omics data sets," *Brief. Bioinform.*, pp. 1–4, 2017.

[25] B. G. Tabachnick and L. S. Fidell, *Using multivariate statistics, 5th ed.* Boston, MA: Allyn & Bacon/Pearson Education, 2007.

[26] S. S. Manual, "Full-Text."

[27] A. A. Miranda, Y. A. Le Borgne, and G. Bontempi, "New routes from minimal approximation error to principal components," *Neural Process. Lett.*, vol. 27, no. 3, pp. 197–207, 2008.

[28] J. Maindonald and W. J. Braun, "Data Analysis and Graphics Using R - an Example-Based Approach (3rd edition)," *Cambridge*, p. 565, 2003.

[29] N. Lunardon, G. Menardi, and N. Torelli, "ROSE : A Package for Binary Imbalanced Learning," *R J.*, vol. 6, no. June, pp. 79–89, 2014.

[30] B. Lantz, *Machine learning with R*. Birmingham, UK: Packt Publishing, 2015.