Exoplanet Detections with Machine Learning

Antonio Gordillo Toledo, Gavin Groode, Jie Qiu, Jai Yarlagadda
University of California Berkeley Physics Department, Undergraduate Lab at Berkeley

Abstract

Telescopes such as the Kepler Space Telescope are collecting vast amounts of data. Manually sifting through these large datasets presents a problem when attempting to find exoplanet candidates within the thousands of generated light curves. Machine Learning (ML) techniques can be implemented to automatically look for patterns correlated to planetary candidates. Before doing so, data must be processed and normalized to be properly input into a ML model. Two different approaches were taken when designing the ML model. Initially, an SVM (support vector machine) implementation was created. After poor results, the ML approach pivoted from SVMs to CNN (Convolutional Neural Networks.) The final CNN yielded an accuracy of ~81.8%. Further modifications to the method used for data processing and refining the CNN model could potentially improve these results.

Kepler Dataset

The Kepler Exoplanet Archive was used to generate a list of Threshold-Crossing Events, along with required metadata. A Threshold-Crossing Event (TCE) is a sequence of transit-like features in the flux time series of a given target that resembles the signature of a transiting planet to a sufficient degree that the target is passed on for further analysis.

The datasets size and labelled TCEs made the Q1-Q17 DR24 Kepler dataset ideal for using with ML. The dataset contains TCEs labelled as PC (Planetary Candidates), AFP (Astrophysical False Positives), NTP (Non-Transit Phenomenon), and UNK (Unknown). TCEs labelled UNK were removed. PCs were relabelled as '1' for planet candidate and those labelled either AFP or NTP were relabelled as '0' for non-planet candidates. The dataset included 15,737 total TCEs. This was composed of 3,600 PCs, 9,596 AFPs, and 2,541 NTPs.

Data Processing

The list of TCEs was then used to obtain the corresponding flux time series data from the Mikulski Archive for Space Telescopes. Each TCEs data was split into various FITS files for the different observation quarters. Most TCEs were broken into 15-18 individual FITS files.

Each FITS file contained two sets of flux time series data, Simple Aperture Photometry (SAP_FLUX) and Pre-search Data Conditioning (PDCSAP_FLUX). The PDCSAP_FLUX flux time series data was extracted from the FITS files because it has undergone an additional step in the Kepler pipeline which has removed long-term trends. **Figure 1** shows a comparison of the two sets of flux time series data corresponding to KIC 6022556. As can be seen, SAP_FLUX demonstrates a long-term negative trend, whereas the PDCSAP_FLUX has been flattened and cleaned of the long-term trend.

Each TCEs FITS files needed to be individually flattened, cleaned, and normalized before being stitched together. The combined flux time series was then folded using the TCEs period, which was obtained from the acquired metadata. The result of this is a cleaned and normalized folded light curve. **Figure 2** shows the first time series data of KIC 6022556 (which is the same data as Figure 1). **Figure 3** shows the various FITS files of KIC 6022556 being stitched together. **Figure 4** shows the resultant folded and cleaned light curve corresponding to KIC 6022556. **Figures 5, 6, and 7** show the folded light curves of TCEs labelled AFP and NTP.

Figure 1

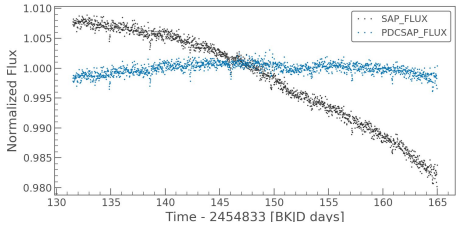


Figure 2

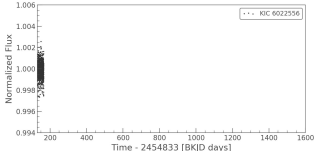


Figure 3

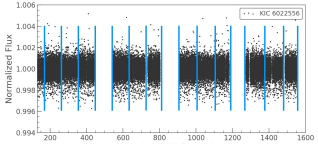


Figure 4

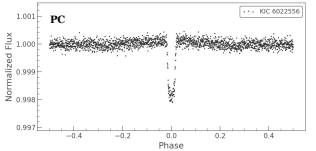


Figure 5

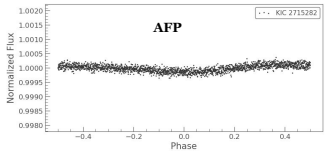


Figure 6

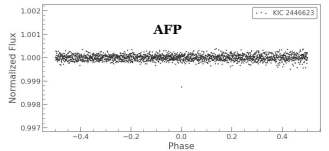
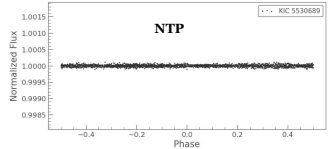


Figure 7



SVM Model

Before diving into Convolutional Neural Networks (CNNs), the problem of classifying interstellar objects initially used Support Vector Machines (SVMs) and Fourier Transforms. Instead of folding light curves, a Fast Fourier Transform was used on flux data in attempt to split light frequencies of exoplanets versus other objects such as binary star systems. However, the accuracy of this method was only ~40%. The low accuracy of this method can be explained by the heavy amount of noise that exists in exoplanet datasets and star flux datasets, a drawback of SVMs. Furthermore, a lack of consistent sinusoidal variations within the datasets could be the culprit to poor performance with Fast Fourier Transforms, as certain Fourier Techniques rely more so on sinusoidal data. Additionally, noise, high levels of gaps in data, and lack of consistency could all play a role in the disadvantage of using Fourier Transforms in this case.

For the SVM, we imported the data and did basic processing such as normalizing. Then, the Fast Fourier Transform method was used on the data, and finally the SVM model was deployed on the data. Afterwards, hyperparameter tuning was conducted, however rarely did tuning make large differences to the accuracy in the end. An interesting point to be made is that drastic changes (such as not normalizing vs normalizing the data or applying filters) did not correlate to large drastic changes in accuracy.

A more fitting pipeline for this task is to replace Fourier Transforms with phase folding techniques, allowing non-sinusoidal and high noise data to be worked on, as well as using a CNN instead of an SVM. Our accuracy greatly improved after changing methods.

CNN Model

In a convolutional neural network, we create multiple layers of "neurons." Each "neuron" in the layer processes only a small part of the layer before it. This allows the network to distinguish shapes and edges through the contrast neighboring neurons detect. Like most neural networks, having multiple layers allows for a more fine-tuned solution, where output from different parts of the image can be analysed together when processed by the additional layers. To make sure all the data is processed, the final layer is densely connected. It is primarily this processing of edge and shape that led us to switch from an SVM to a CNN. This edge detection helps the CNN detect the dip in the light intensity with much more accuracy than the SVM could.

Initially, we tried to to adapt a 2D image classification model with two convolutional layers, two pooling layers, and a fully connected layer. Since our input data are 1D, we first replaced all conv2D functions with conv1D, we then tweaked the batch size, stride, and the layer dimensions to accommodate our input data size, which consists of 2001 points. We also dropped the max pooling layers. Our final model has a fully connected layer as well as two convolutional layers with 32 and 64 output channels, respectively. However, accuracy did not improve as we trained our data with this model.

A previous research paper outlined a successful model in similar application and the approach was shifted to emulating a similar model (Shallue 2018). The paper referred to a CNN model that received two views of the light curves as input; a global view and a local view of the light curves would help distinguish long-term and short-term patterns in the data. As for the model itself, a simplified version was created to get an initial training run. The model went through various iterations before arriving at something similar in structure to the one outlines in the paper. The model adapted from the 2D Image Classifier reached an accuracy of 75.73% and the model based off the paper improved accuracy to 78.72%. In addition to the convolution, pooling, and dense layers, it was found that adding dropout layers helped increase both the accuracy and the time it took to reach the highest accuracy, which was 81.81% at the end of the project. This maximum accuracy did not emulate the ones in the paper. The step in the process that would most benefit from revising would likely be data processing.

Future Work

Accuracy of the CNN model could be improved through various modifications to both the data cleaning process and CNN model itself. The period of each TCE (used to fold the flux time series data) was obtained from the metadata provided from the Exoplanet Archive but manually finding each TCEs period through a more rigorous method could improve accuracy. The processing time of each TCE would increase but yield a better folded light curve. Another potentially helpful change would be to remove the data of multiple planet candidate transits from each TCE, as many TCE light curves contain more than one transit. In addition to this, creating another view of the light curves could help find different patterns that are missed by inputting a single light curve view into the model. For every TCE, only one global view light curve was generated, but adding a 'zoomed in' local view might help detect otherwise easily missed patterns. One final change would be to add an additional class to the labels, so that the model could be used to distinguish PCs, AFPS, and NTP. The current model makes no distinction between AFPs and NTPs.

References

Ansdell, M., Ioannou, Y., Osborn, H. P., Sasdelli, M., Smith, J. C., ... Caldwell, D. (2018). Scientific Domain Knowledge Improves Exoplanet Transit Classification with Deep Learning. The Astrophysical Journal, 869(1), L7. https://doi.org/10.3847/2041-8213/aaf2ab

Mikulski Archive for Space Telescopes http://archive.stsci.edu/kepler/
NASA Exoplanet Archive. Caltech. https://exoplanetarchive.ipac.caltech.edu.

Shallue, C. J., & Vanderburg, A. (2018). Identifying Exoplanets with Deep Learning: A Five-planet Resonant Chain around Kepler-80 and an Eighth Planet around Kepler-90. *The Astronomical Journal*, 155(2), 94.

Seader, S., Jenkins, J. M., Tenenbaum, P., Twicken, J. D., Smith, J. C., Morris, R., ... Klaus, T. C. (2015). DETECTION OF POTENTIAL TRANSIT SIGNALS IN 17 QUARTERS OF KEPLER MISSION DATA. The Astrophysical Journal Supplement Series, 217(1), 18. https://doi.org/10.1088/0067-0049/217/1/18

Acknowledgements

We would like to thank Lauren Koch and Tristen Streichenberger for their support throughout this project, as well as Arjun Savel and Megan Ansdell.