Searching for Exoplanets with Machine Learning

Eli Wiston COL ‘22, Sarah Kane COL ‘23, Cyrille Doux, Professor Cullen Blake, Professor Bhuvnesh Jain
University of Pennsylvania School of Arts and Sciences, Department of Physics and Astronomy

Penn

Background:

- We used stellar light curves from NASA’s **Transiting Exoplanet Survey Satellite (TESS)** mission in our analysis. We were looking for signals from transiting exoplanets that cause periodic dips in the light curve, the plot of brightness versus time, of their host star.
- There are thousands of TESS light curves, and the signals from transiting exoplanets are tiny.
  - It’s like looking for the needle in the proverbial haystack!
- **Convolutional neural networks (CNNs)** are machine learning algorithms that can:
  - Categorize data by the presence of features.
  - Make predictions on many samples in a short time.

GOAL: Can we identify new exoplanet candidates by using a convolutional neural network to search for transits in stellar light curves?

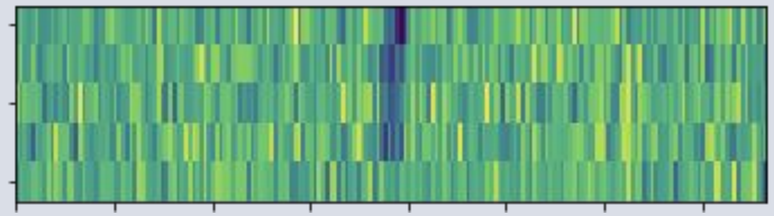
Data Preparation

- **1D Phase Fold:** The transit least squares (tls) algorithm is used to find the best estimate of transit parameters. The light curve is then phase folded on that period, amplifying the signal of real transits.
- **2D Data Folding:** Timeseries light curve is cut and reshaped to make a 2D “image” that emphasizes the transit signal for network detection. See Figure 1 for method details.
- **1D Wavelet Transform**: The “Mexican hat” wavelet transform is applied to the light curves at a variety of frequencies. We then compute statistics (power, mean, standard deviation, etc.) for each transformed curve and analyze these values in a 1D CNN
- **2D Wavelet Transform:** Wavelet transforms are computed in the same way as above; however, now we pass each wavelet transformed curve into the network.

Results

Both the 1D phase fold and 1D wavelet transform methods were effective in simple simulations, but failed to generalize to more realistic data. The 2D wavelet transform shows promise, but poses some computational challenges with system memory. The **2D data folding approach has been effective with Lilith data**, with the results shown in Figure 2.

With simulated exoplanet transit (strong signal):



Without simulated exoplanet transit:

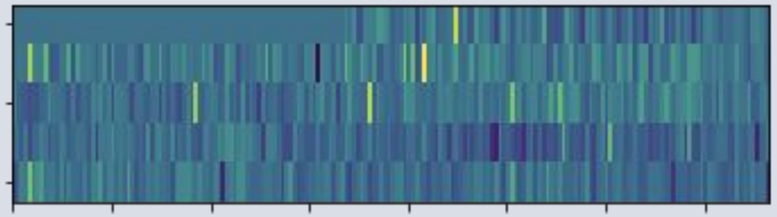


Figure 1 - 2D folded light curve with simulated transit (top) and without transit (bottom). Light curves are cut into 4 d segments with transits centered, and those segments are subsequently combined to make the “images” above. The transit, visible as the dark line in the center of the top image, is more easily identifiable by the neural network in this form.

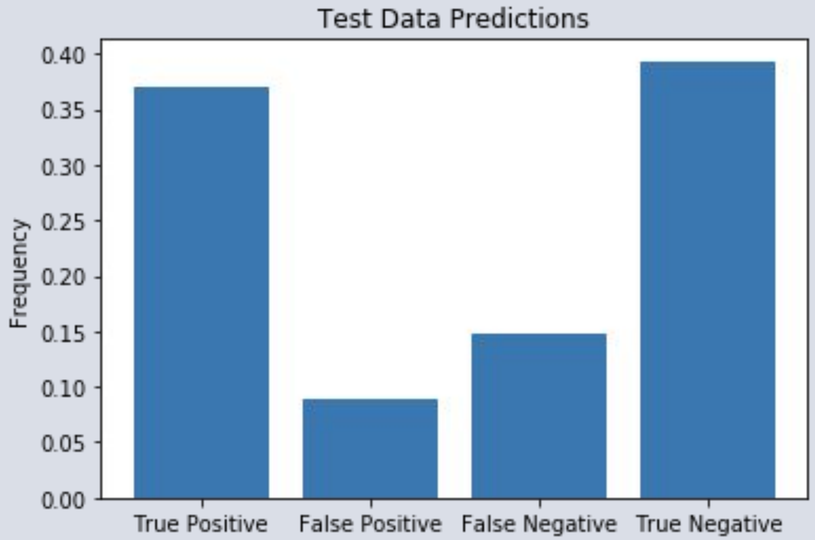


Figure 2 - Network predictions for Lilith data with inserted transits of 4 - 7 Earth radii orbiting a solar analog. Data prepared using the 2D folding method (see Figure 1). Overall network accuracy at this signal strength is approximately 75%. Accuracy improves significantly with higher signal-to-noise transits.

Methods:

1a. Make **simulated light curves** with simple Gaussian noise and inserted transits created by BATMAN, an exoplanet transit simulation package for Python.

1b. **Create neural network** architecture using Tensorflow, a Python package for machine learning made by Google.

2. **Prepare the data** using one of several tested methods:
- 1D phase folding
- 2D data folding (see Figure 1)
- 1D wavelet transform
- 2D wavelet transform

3. **Train the network** on the prepared data. **Test** network accuracy on exoplanet transits of varying signal-to-noise ratios. Adjust the network architecture and the processing methods as needed.

4. Repeat Steps 2 and 3 using light curves from **Lilith, highly accurate simulated data** provided by NASA. Test using CNNs from Step 1b.