# Searching for Earthlike Worlds: Detection of Habitable Small Planet Candidates in Kepler Data with a Deep Neural Network

Claire T. Chen

Regeneron Science Talent Search 2020

November 8, 2020

# ABSTRACT

Habitable small planets (HSPs) are a uniquely interesting population of exoplanets. Defined as terrestrial planets roughly of Earth's size, orbiting in their star's habitable zone, HSPs provide a window to better understanding the development of planets in general and Earth-like worlds in particular, as well as helping place the solar system within the broader context of the Milky Way, and guiding the search for life outside of Earth. Currently, there are fewer than 60 confirmed HSPs, making more comprehensive population studies difficult. This paper presents a set of new and enhanced methodologies to facilitate more efficient discovery of HSP candidates in targets from the Kepler catalog. They include a windowed fastfolding method that leverages GPU parallel processing to quickly process lightcurve data, an enhanced method for simulating HSP transit signals to rapidly generate a large dataset of positive samples for neural network training, and a deep neural network (DNN) trained with the population of synthetic samples and demonstrated to be able to successfully recover 11 out of 11 known HSPs with periods in the 100-day to 200-day range with high confidence. Focusing on identifying potential small planets orbiting F-, G-, and K-dwarfs with periods between 100 and 200 days, these methodologies were then applied to process and analyze lightcurve data from 148 Kepler targets, resulting in the discovery of 6 new HSP candidates.

# 1. Introduction

Ever since the first detection of extrasolar planets orbiting a main sequence star in 1995, the hunt has been on for new worlds throughout the Milky Way. One particularly interesting population of exoplanets is that of habitable small planets (HSPs), terrestrial planets roughly of Earth's size with the potential to support the development of life. More specifically, HSPs are defined as planets with radii smaller than 2.0 times Earth's radius, with orbits within the "habitable zone", the range of distances around a star in which an orbiting planet can be capable of sustaining surface-level liquid water, the fundamental prerequisite for life as we know it. The habitable zone can be divided into two regions: a narrow "conservative" habitable zone where planets like Earth can maintain a warm enough surface temperature to allow for liquid water, and a wider "optimistic" habitable zone in which a planet like Venus or Mars could conceivably support surface-level water given adequate atmospheric conditions. The specific size of the habitable zone depends on the flux of the star: it would extend farther out for brighter, hotter stars, and be closer in for dimmer, smaller ones. The inner edge of the optimistic habitable zone, for example, can include planets with orbital periods from around 100 days to over 400 days for K- and F-type stars respectively, with the conservative habitable zone's inner edge covering periods in the 200-days for K-type stars, going up to over 600 days for F-type stars (Kopparapu et al. 2013).

Studying HSPs and habitable zones provide a fascinating opportunity to explore important questions about the occurrence, development, and evolution of planets like Earth. The discovery of more HSPs will help place Earth and the solar system within the broader context of star systems in general, and in the Milky Way in particular: is Earth special, or are there many more Earth-like planets orbiting in similar systems throughout the galaxy? In addition, learning about these potentially habitable worlds can guide the search for life outside of Earth, providing possible targets for further study. To date, though, fewer than 60 habitable small planets have been confirmed, out of over 4,281 confirmed exoplanets and a growing list of planet candidates. One major source of these planet discoveries is in data from the Kepler mission, which conducted photometric observations of over 530,000 stars in its missions from 2008 to 2018. This data is accessible in the form of lightcurves, which are graphs of the amount of light produced by a star over a period of time. The hunt for potential exoplanets typically begins with identifying threshold crossing events, periodic "dips" in the lightcurve that could be produced by a transiting planet. When these periodic events are found, they are carefully examined to eliminate apparent signals caused by noise, eclipsing binaries, or stellar activity, a process done manually by humans. However, if one is interested in studying some certain population of planets, the limitations of this process become clear as it is inefficient to find large numbers of candidates, which is a strong motivator for the development of a more efficient method for identifying potential transiting planets.

HSPs' longer orbital periods, coupled with their small size, mean their transits will be relatively shallow, wide, and thus difficult to detect. In fact, it is possible that the Kepler catalog is only 50.5% reliable for longer-period planets per Thompson et al. (2018), which means that the current sampling could be missing half the population of yet-undiscovered HSPs. Another challenge for detecting HSPs is that even within the set of confirmed HSPs, properties can vary significantly -- as shown in Figure 1, orbital periods range from 8.689 days to 395.113 days, with a distinct large group of planets with sub-100-day periods. Due to the variety in properties, it would be more efficient for a study to focus on a narrowed period range, to be able to better detect and analyse a specific subgroup of HSPs. For this work, then, the population of planets to search for was defined as HSPs with orbital periods in the range from 100 days to 200 days, covering the inner edge of the habitable zone for main-sequence stars.

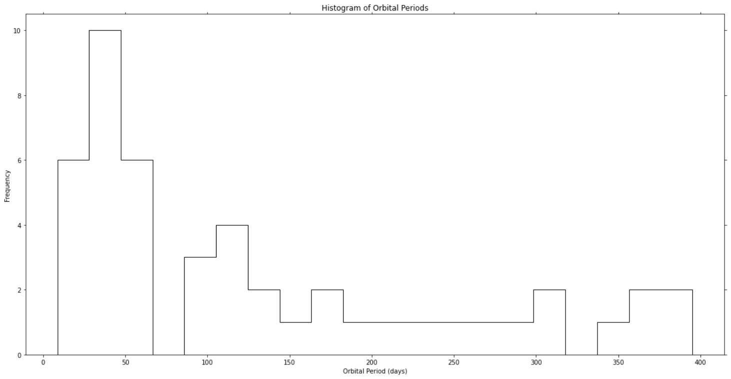


Figure 1: The distribution of orbital periods, in days, of known HSPs. The spread of the periods is large -- from a minimum of 8.689 days to a maximum of 395.131 days -- and the range of properties of these planets make it difficult to study as one group.

This paper develops a set of new methodologies which improve the efficiency of discovering HSP candidates and uses them to identify 6 new HSP candidates. The paper is organized as follows. First, Section 2 provides an introduction to convolutional neural networks and their particular application. Section 3 will cover the preprocessing of raw lightcurve data and present the new method of windowed fastfolding, which prepares data for inputting into a neural network model. Section 4 describes the process for the generation of simulated transit data for training and testing the model, as the amount of real HSP data is too limited to effectively train on real data. Section 5 discusses the architecture and structure of the neural network model itself and the training procedure. In Section 6, the model's performance on testing with synthetic data is examined, and the postprocessing necessary to ensure the recovery of transits is discussed. Results in Section 6 also verify the performance of the neural network model by confirming known transits in real lightcurves with high confidence. Next, Section 7 presents the results of the initial search in the Kepler catalog using this full process: new habitable small planet candidates and their properties, as fitted and calculated with the Markov chain Monte Carlo (MCMC) method. Finally, in Section 8, the prospects of future exploration with this work are examined.

## 2. An Introduction to Convolutional Neural Networks

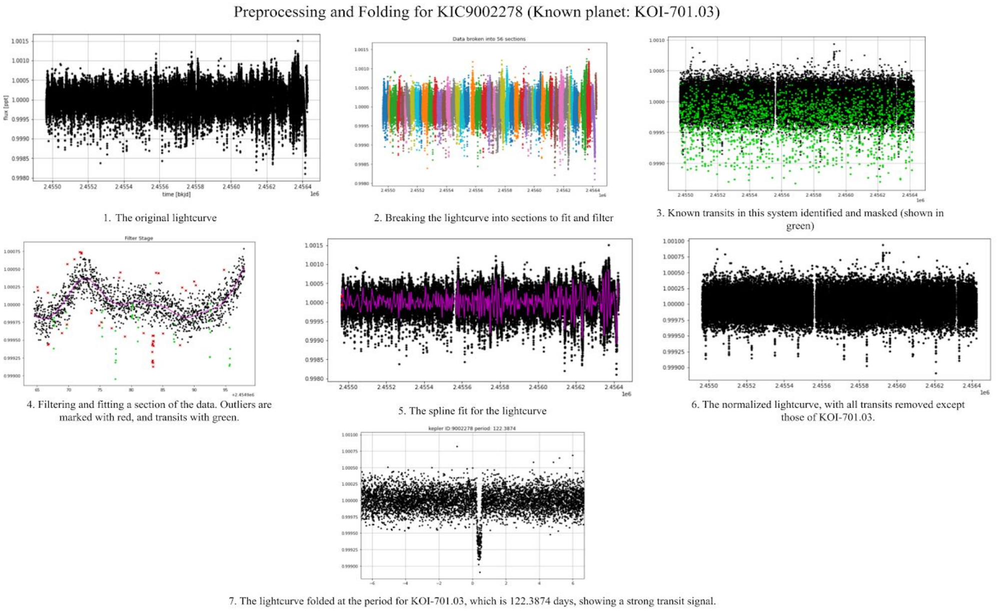
The recent development of neural networks has proven to be very promising in approaching problems involving classification and finding patterns. At a basic level, a neural network consists of multiple layers of nodes, or neurons, which are densely connected with each other within the layers, all sending and receiving data. Weights and biases, the learnable parameters of the model, define the strength of connections between the neurons and guarantee activity of the neurons, determining how the neuron “fires” and assigning importance to different features in inputs. Although each neuron is a simple element, the connections and layers formed by many neurons can process information, make decisions, and adapt responses to external factors. In a neural network, these algorithms behave in a way analogous to neurons in a brain to identify complex patterns that a human might miss, and crucially, judge and learn from its own performance and adapt to improve. One specific kind of neural network is a convolutional neural network, or CNN. Its distinguishing feature is the use of convolution layers central to its operation. These layers are so-named because they perform the operation of convolution, where a filter, also known as a kernel, is applied to an input by multiplying it with a set of weights. A single convolution layer can have multiple filters, which when applied across the range of an inputted array of data, results in a “feature map” describing the locations and significances of features detected in the input. The feature map can then be normalized and be passed through an activation function, which defines the output of the neuron -- how the neuron “fires”. CNNs are commonly used in image-processing applications to aid in extracting patterns or features present in an image, due to the fact that convolution layers allow for the learning and interpretation of features across an entire input. The same principle can be applied with regards to searching for planet transits, as transits represented by adjacent flux values forming a “dip” pattern are features that need to be learned across the range of inputted data points. Since this is something CNNs excel at, they can be a powerful tool for identifying the transits of potential new HSP candidates more efficiently.

## 3. Input Data and Preprocessing

This work uses data originating from the Kepler mission, downloaded over all available quarters from the Mikulski Archive for Space Telescopes and stitched together to create full lightcurves. Each quarter represents approximately 90 days’ worth of data containing, on average, 3,900 points evenly spaced in approximately 33.23-minute intervals, and with around 17 or fewer quarters of data available for each target, the resulting lightcurves have between approximately 50,000 and 70,000 data points representing up to around four years. As the periods of the planets being searched for are between 100 and 200 days long, it is expected that there may not be many transits present in the lightcurve: at most, for a planet with a 100-day period, the data would cover less than 15 periods, and if the planet had a period of 200 days, there may be as few as 7 periods represented. So, it is essential to further prepare and process the lightcurve to increase the likelihood of successfully recovering any of these longer-period planets’ transits, before it can be analysed by the neural network.

The first step of preprocessing was to normalize the lightcurve. Any points corresponding to the transits of known planets in the lightcurve were found and masked out, as were significant outliers, so as not to impact fitting of the data. Then, the lightcurve was broken into smaller sections and fit with a cubic smoothing spline. The flux values were then divided by the spline fit to detrend the data, removing low-frequency variability.

Àfter normalizing and filtering, the next step for generating neural network inputs is to fold the lightcurve along trial periods and bin the data. Folding can be thought of as taking sections of the lightcurve, each representing a length of time equal to the trial period, and "stacking" them. If a transit with this period is present, the signals add up, resulting in a stronger signal. If the trial period does not match the period of any transit in the data, the stacked signals end up cancelling out. Folding at the right period brings out signals, even weaker ones, that can then be analyzed by the neural network.



1. The original lightcurve

2. Breaking the lightcurve into sections to fit and filter

3. Known transits in this system identified and masked (shown in green)

4. Filtering and fitting a section of the data. Outliers are marked with red, and transits with green.

5. The spline fit for the lightcurve

6. The normalized lightcurve, with all transits removed except those of KOI-701.03.

7. The lightcurve folded at the period for KOI-701.03, which is 122.3874 days, showing a strong transit signal.

Figure 2: The process for preparing a lightcurve, step by step, using KIC9002278 as an example. The last plot is the folded transit for known planet KOI-701.03 using the period in the Kepler catalog.

In the absence of a known transit period, a trial period for folding needs to be determined. One commonly used method for modeling and calculating the parameters of transits is with the Box Least Squares algorithm (BLS, Kovács et al. 2002), in which transits are modeled as periodic "boxes" with height corresponding to signal depth and width corresponding to signal duration. The result of this process is a power spectrum over a given range of trial periods, in which a peak corresponds to the model's best prediction for the transit period. With BLS, an estimate for the period and transit  reference time can be determined, but the method has its limitations. A major weakness of the BLS algorithm is that shallow transits and longer-period transits often do not produce enough power in a power spectrum to appear as peaks and thus be detected as potential candidates. The transits of HSPs tend to be wider and shallow, with longer periods, so they may be overlooked by the BLS algorithm. Because of this, a new alternative method was developed, where a lightcurve would be folded on a given number of periods in a given range, eliminating the need for calculating a specific trial period.

This new procedure, windowed fastfolding, could significantly reduce the time needed to prepare lightcurves for input into a neural network model. Given a range of periods $[p_1,\ p_2]$ to attempt and *x*, some number of periods to attempt, a given lightcurve will be folded $n$ times, at periods starting at $p_1$ and incrementing by $\frac{p_2 - p_1}{x}$ days each time. The output for each period $p$ is a “window” of fixed size $n$, where each point represents a time unit of $\frac{p}{n}$ days or $\frac{p}{24n}$ hours. In other words, windowed fastfolding is a method to quickly "brute force" fold a lightcurve, if a range of periods to try and the resolution of the desired result is known. The program to execute this process is specifically written to be run on an NVIDIA GEFORCE RTX 2080 GPU, allowing for the use of GPU parallel processing to improve efficiency. Unlike the standard method of manually folding a lightcurve with a period found by BLS, windowed fastfolding does not have the prerequisite of knowing a specific value for the period. Since one does not need to search for a period, this reduces the time needed to fold each target -- finding a period by running the BLS algorithm takes more time. In fact, on a timed trial of fastfolding a lightcurve 44,000 times with a range of periods from 100 days to 200 days, it took 11.91 seconds for the folding to be completed and an additional 7.89 seconds to save the file containing all the results for a total of 19.80 seconds. In contrast, if the current method of finding a period with BLS and manually folding was applied with the same period range and level of precision, it took 22.87 seconds to calculate just one period. Considering that this was just the time needed to determine one possible value for the period, it would take even more time to manually fold all of the potential transit signals with different periods to eventually pick out candidates for planets. Windowed fastfolding allows for a large number of folds to be generated and ready to feed in in less time, the idea being that the way to improve the chance of recovering transit candidates is to look at as much data as possible. It also has the advantage over BLS in that it is more sensitive to shallower and longer-period signals compared to BLS, as can be seen in Figure 3. A transit feature is apparent in the fastfolded result, shown in the top plot. Compared to the manually folded result in the bottom plot, the fastfolded signal is stronger, despite the same bin number used in both procedures.

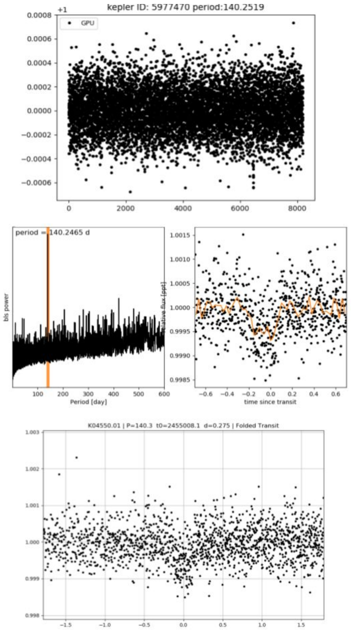


Figure 3: Fastfolding result (top) compared with folding manually (bottom) with the period as calculated with BLS (center).

With the proposed fastfolding program, the final output is always a fixed size of 8,192, no matter the period, meaning that each folded result represents a different amount of time with the same number of total points. The reason the fixed window size is 8,192s is because using fewer points runs the risk of losing shorter-duration transits. If the minimum period in the range to search is 100 days, each point in the fastfolding program’s output would represent 0.0122 days, or 0.293 hours. If however the lightcurve is folded at a period of 200 days, each point in the output would represent 0.0244 days, or 0.586 hours. Because of this, in the outputs of the fastfolding process, transits could appear as only a few points. From the distribution of transit durations for known HSPs, the shortest duration was around two hours while the longest was above 20 hours. A 20-hour transit would be represented as 68 points (if the period was 100 days) or 34 points (if the period was 200 days), while a two-hour transit would be represented as few as six points (if the period was 100 days) or three points (if the period was 200 days). A signal represented by only a couple of points in a size 8,192 window would be practically undetectable -- or completely disappear -- if the window size was decreased, but a small cluster of points could still be a prominent enough feature for a neural network to identify. Thus, the window size of 8,192 was selected as it was the minimum size where folding with any period in this target range would not run a significant risk of losing short transits.

## 4. Creating Training Data

Training of a neural network model is the process of tuning and adjusting its weights to accurately detect the desired features. To turn a poorly-performing neural network into a high-performing one, the model needs to be trained with a large sample of data containing the desired feature and having it automatically infer rules for recognizing the feature. The more samples there are to train on, the more the network can learn and improve its rules -- training sets often contain thousands of samples, if not hundreds of thousands or more. However, for HSPs, their small population does not provide enough data to make a training set of real data. One solution is to simulate transit signals with a simple shape to closely approximate the properties of real HSP transits. This would ensure a sufficiently large sample of signals that can be used to train the neural network. Given the parameters of transit depth, duration, period, and the ratio of planet radius to star radius, trapezoids were generated to roughly represent transit signals. The trapezoid was then injected into a range of times representing the period time window, at a random location to account for the fact that a transit could be present anywhere in the window, including at either end. Then, Gaussian noise was added, simulating the presence of noise and artifacts present in real lightcurves. As the neural network would eventually have to detect real transit signals, the artificial data generated had to account for the entire range of possible properties that a real HSP transit could have. Looking at the distribution of durations for HSP transits, a transit's duration could be anywhere from around 2 hours to around 21 hours. Since fastfolding produces a window representing a period of time anywhere from 100 days to 200 days, then a transit's duration as shown in the folded lightcurve would fall between 0.000415 times the length of the period to 0.0083 times the length of the period; thus the durations of the synthetic transits were generated to fall in the range from $0.000415 \times 100$ days to $0.0083 \times 200$ days. As the depths of real transits ranged from $7.2199 \times 10^{-5}$ to 0.0026046, the depths of the synthetic transits would be generated such that they comfortably included this interval, ranging from a minimum depth of $3.2037 \times 10^{-6}$ to a maximum depth of 0.00620.

Figure 4 shows the result of this process: a generated trapezoid transit signal, which was then randomized. The "clean" trapezoid can be seen as the red line in the plots on the right. The signals were injected into a flat base at a random location as a folded transit's time of reference can be present anywhere in the window. In the same figure, the plots on the left show the final artificial sample, generated by adding Gaussian noise within a previously-defined range to simulate noise in real lightcurve data.

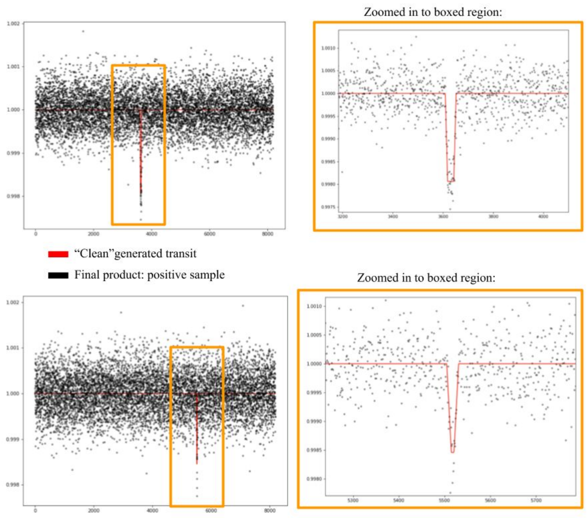


Figure 4: 2 examples of artificial transit signals generated using the trapezoid method.

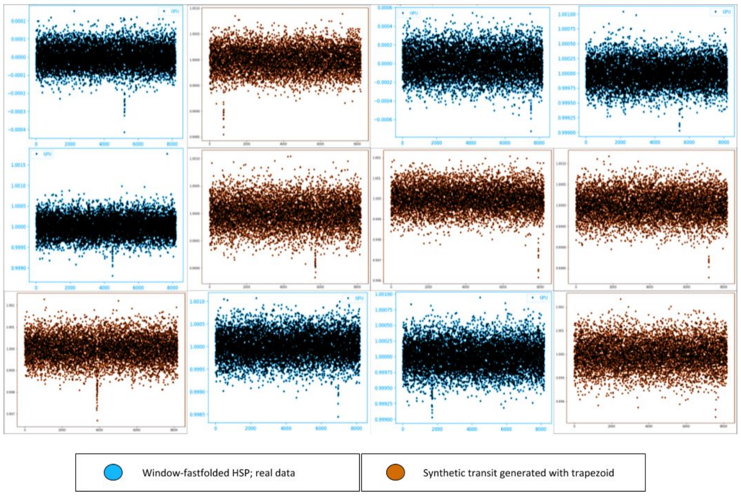
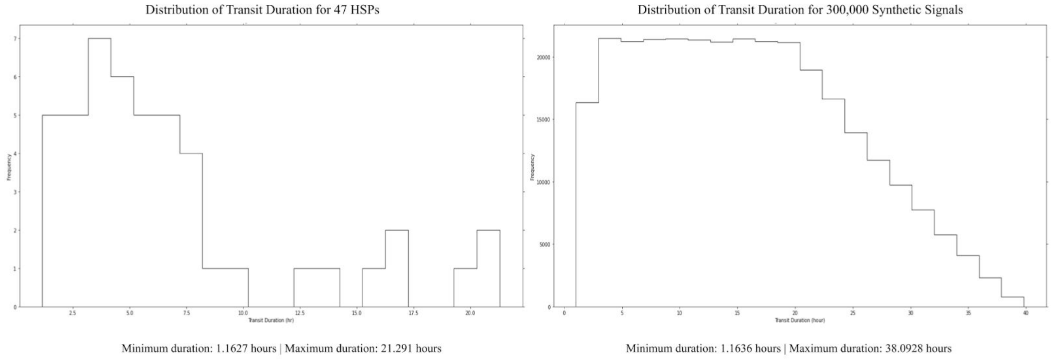


Figure 5: A comparison of real fastfolded lightcurve data with planet transits and artificial positive samples generated with the trapezoid method.


Distribution of Transit Duration for 47 HSPs

Distribution of Transit Duration for 300,000 Synthetic Signals

Minimum duration: 1.1627 hours | Maximum duration: 21.291 hours

Minimum duration: 1.1636 hours | Maximum duration: 38.0928 hours

Figure 6: Comparing the distribution of transit durations, in hours, of 47 HSPs (left) with those of 300,000 artificial transits generated for the training dataset (right).

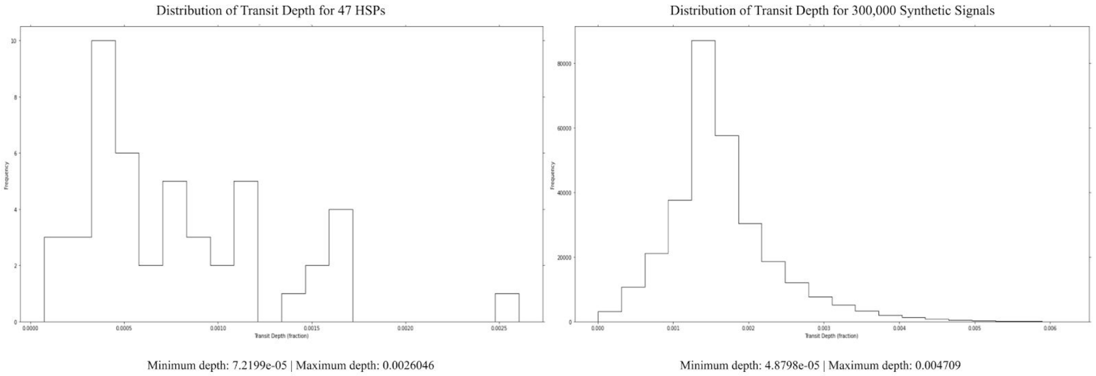


Figure 7: Comparing the distribution of transit depths, of 47 HSPs (left) with those of 300,000 artificial transits generated for the training dataset (right).

One quick check for the quality of generated artificial data for the training set is to visually compare synthetic positive signals to transits in real fastfolded lightcurves. As shown in Figure 5, the real and artificial signals are indistinguishable by eye, both showing very similar features. Another test for the synthetic data is to compare the distribution of properties of the generated transits with those of real planets' transits in order to ensure that the synthetic data fully accounts for the range of features which the neural network will need to identify. Comparing the distributions of transit durations (Figure 6) and depths (Figure 7) for synthetic and real transits side-by-side, the range of durations and depths in the artificial data fully covers the range of durations and depths found in the real data, so it appears to be a fairly representative simulation.

## 5. The Neural Network Model

### 5.1 Architecture

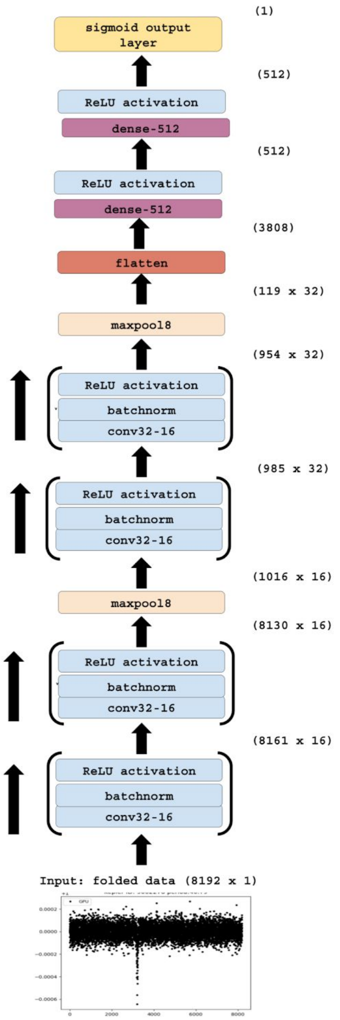


Figure 8: The architecture of the neural network model.

This convolutional neural network has a structure consisting of two pairs of convolutional layers. Each convolution layer has 16 filters and a kernel size of 32, and is followed by a batch normalization layer and a ReLU activation function. The batch normalization layers scale the data output from the convolution layer, standardizing its activations and helping stabilize and regularize the training process, and the activation function contributes some complexity to the neural network to allow it to better recognize any patterns present. Every other convolutional layer is followed by a max pooling function with a pool size of 8. At this operation, the local maximum of subparts of the inputted data is calculated and the data is "pooled", accumulating the data to highlight the most prominent feature within it, which serves to abstract the data and prevent overfitting. After these two sets, the data is flattened. The model was completed with two 512-neuron dense layers, each followed by a ReLU activation function, and a final sigmoid output layer.

One consideration for the model's design is the size of the datasets used for training. Each individual sample is large -- 8192 points -- and thus the largest training set that could fit in memory had only 100,000 total samples. Keeping this in mind, randomly dropping out or ignoring data thus might cause problems by "thinning" out the neural network while it is training. Because of this, there are no dropout layers used in this model. As dropout layers are usually used to reduce overfitting and generalization error, something else would have to accomplish that role in this model, which is one of the reasons for the two max pooling functions. A diagram of the full neural network model can be seen in Figure 8.

### *5.2 Training*

The training phase involved an Adam optimization algorithm with a learning rate of 0.000001. The model was trained for 100 epochs, with a batch size of 32, on a dataset composed of 100,000 synthetic samples: 50,000 positives (has transit signal) and 50,000 negatives (has no transit signal). The metrics of this stage, plotted against the epoch number, are shown in the two plots in Figure 9. The results of this network were satisfactory considering the smaller dataset, as the training and validation accuracy both began to approach 100 percent, with a final and maximum training accuracy of 99.04 percent and a maximum achieved validation accuracy of 97.1 percent. However, there is a high level of fluctuation in the validation accuracy: over 10 percent at times, as the final validation accuracy of the model dropped to 83.22 percent. The most likely explanation for this result is that the learning rate was too high, causing the algorithm to prematurely "jump out" of some local minima and resulting in missed solutions. To an extent, fluctuations in the validation accuracy could also originate from oscillations that occur when the convolutional layers are incapable of recognizing all characteristics of the transit signals and consequently will consistently and cyclically change a certain parameter in failed attempts to optimize the loss function. Some methods to mitigate this could include adding more convolutional layers, or increasing the size of each convolutional layer. Still, the neural network achieved sufficiently high training and validation accuracy scores, and as shown in the following section, was able to identify positive signals with a high rate of success.

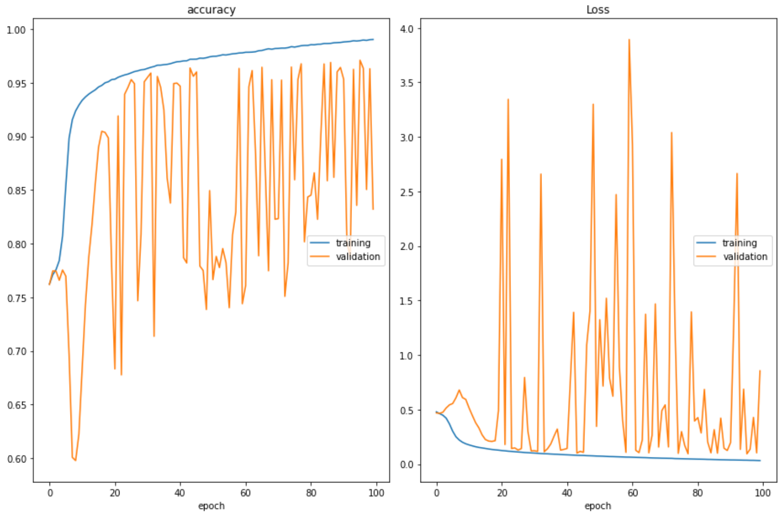


Figure 9: The training phase involved an Adam optimizer with a learning rate of 0.000001. The training accuracy was high, with a final value of 99.04 percent. The validation accuracy trended upwards and was able to reach above 97 percent, but saw heavy fluctuations.

## 6. Testing and Validation of the Neural Network Model

### *6.1 Testing and the Need for Postprocessing*

After training the neural network model on synthetic data, the model was tested with 50,000 artificial samples to evaluate its performance. The overall results were promising: the model was able to correctly identify the samples as either having a transit signal or having no signal 97.18 percent of the time, at a high precision of 99.10 percent as well. Looking at the success rate specifically for generated positive signals, out of 25,000, 23,808 were correctly identified for an accuracy of 95.23 percent. Out of 25,000 negative samples, where no generated transit was present, 24,783 were correctly identified for an accuracy of 99.13 percent (Figure 10).

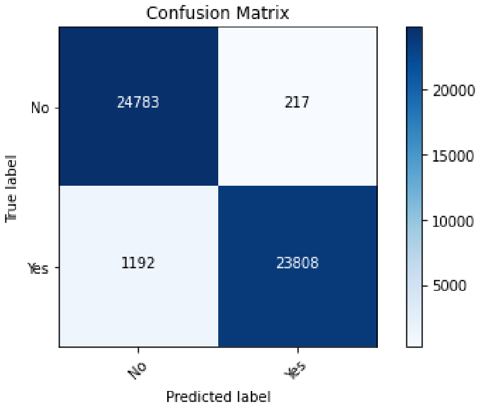


Figure 10: The confusion matrix resulting from testing 50,000 artificially generated samples with the model.

However, it is important to examine where and how the model failed, for the 4.78 percent of generated transits that were not recovered and the 0.868 percent of negative samples in which a transit signal was wrongly identified. As shown in Figure 11, many false negatives originated from short-duration generated signals, which appeared as features consisting of less than 10 points. The way to remedy this would be to generate a larger number of weaker and shorter-duration signals for the training set, further adjusting the distribution of synthetic duration and depth to better reflect the real distributions. Indeed, although the process for generating artificial signals simulates the full range of real transits' durations and depths, a significant amount of the synthetic data has longer durations and deeper transits than that of real signals, as the median synthetic signal duration is 15.487 hours while the median real signal duration is 5.506 hours, and the median synthetic signal depth is 0.001668 while the median real signal depth is 0.0006144.

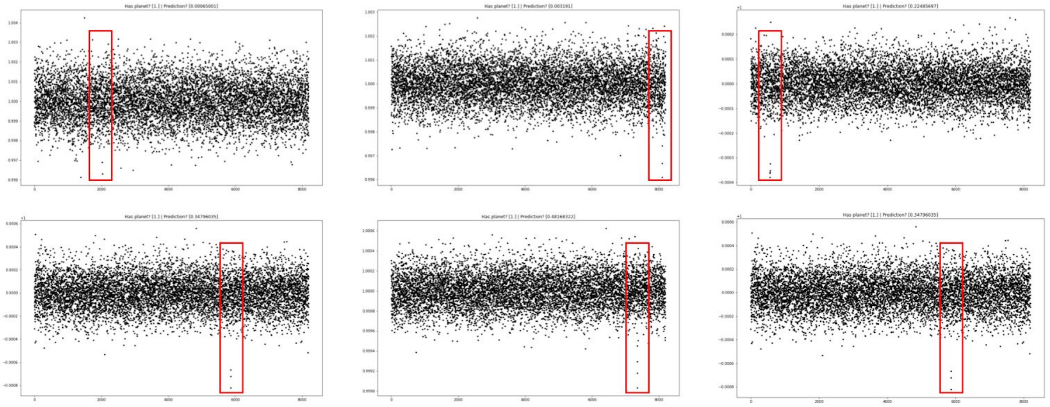
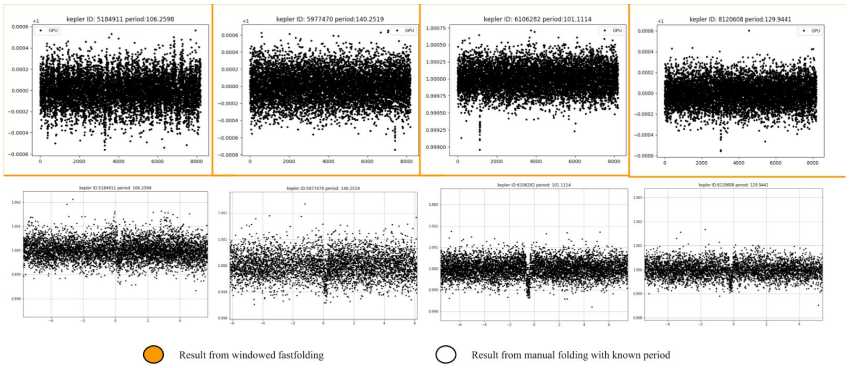


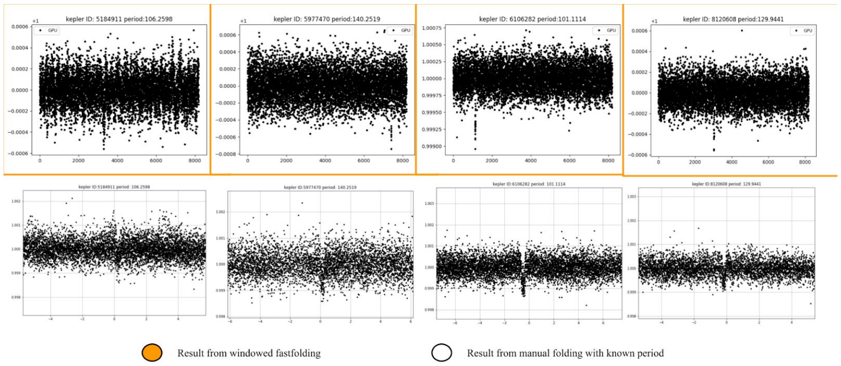
Figure 11: These samples were generated with a transit signal, but were identified as negatives by the model. The location of the generated transit is marked with a red box in each model.

One relevant observation regarding the model's results is that the scaling of confidence scores is not constant -- rather, the scores seemed to scale with noise level. For samples with a high level of noise, the confidence scores tended to be much higher on average, while samples with a low noise level tended to score lower even if features were present. This resulted in some low-noise positive samples falling short of a high-confidence positive recovery by the model. However, this problem can be resolved with the application of postprocessing and rescaling. For several targets, significant spikes were present in the plotted confidence scores as outputted by the neural network model; the returned result only identified the tallest of those peaks and its corresponding period, resulting in cases where the wrong period was identified. If several peaks of similarly high scores appear at different period values, the model returns the one at the greatest period value (as the peak is found moving from low to high on the range of periods). So, after the model processes a target, all peaks in the confidence scores with a significance of $5\sigma$ or higher are identified and returned. This allows local maxima in the scores to be examined and reduces the risk of a transit signal being overlooked. However, in the case of very noisy data with frequent peaks, this process might identify all the noise and produce misleading results, thus, if more than 50 peaks are identified, the postprocessing discards them and returns only the global peak.

### *6.2 Validating with Known Planets*

The final test of the full procedure and especially the neural network model before the search for new candidates can begin is to confirm the detection of known HSPs. Specifically, the model had to demonstrate its ability to successfully identify the transits of small planets in the 100- to 200-day period range by recovering 11 HSPs from the Kepler catalog with periods in this range. The new methodology was shown to be effective in identifying all 11 of these planets. First, windowed fastfolding was shown to be capable of recovering transits for all 11 HSPs; compared to manually folding with the same bin number of 8,192 on the same period value, the transit signals were much stronger and more prominent in the fastfolded results (Figure 12). Running the fastfolded results through the neural network model (Figure 13), the model was able to determine the period for eight out of the 11 HSPs correctly, matching the published period values to within a margin of 0.01 days. However, the recovered highest-scoring periods for the remaining three HSPs did not match the known periods. Instead, these targets had multiple approximately-equal-height peaks in the model confidence scores, originating from periodic variability in the host star. In these cases, the fold at the real period score actually did score highly, but another similarly-scoring peak period was selected as the "best". After postprocessing was applied to recover not just one peak but all the relatively high-scoring indices though, a period value closer to those reported by past studies and the Kepler catalog for these planets was retrieved, confirming that the HSP transits were indeed detected.





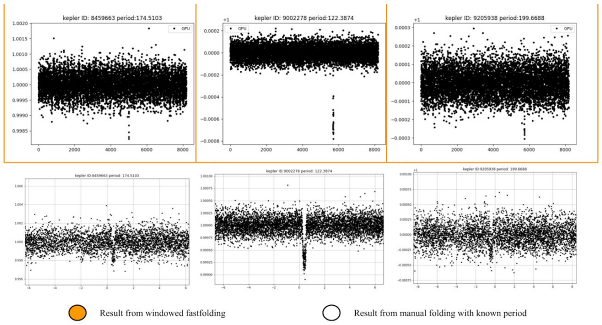


Figure 12: A comparison of fastfolded and manually-folded  transits of 11 HSPs in the 100-200 day period range. A bin size of 8,192 was used for both processes. In the results from windowed fastfolding, outlined in orange, transits appeared as strong but narrow column-like signals, while the signals found with manual folding tended to appear wider and more shallow.

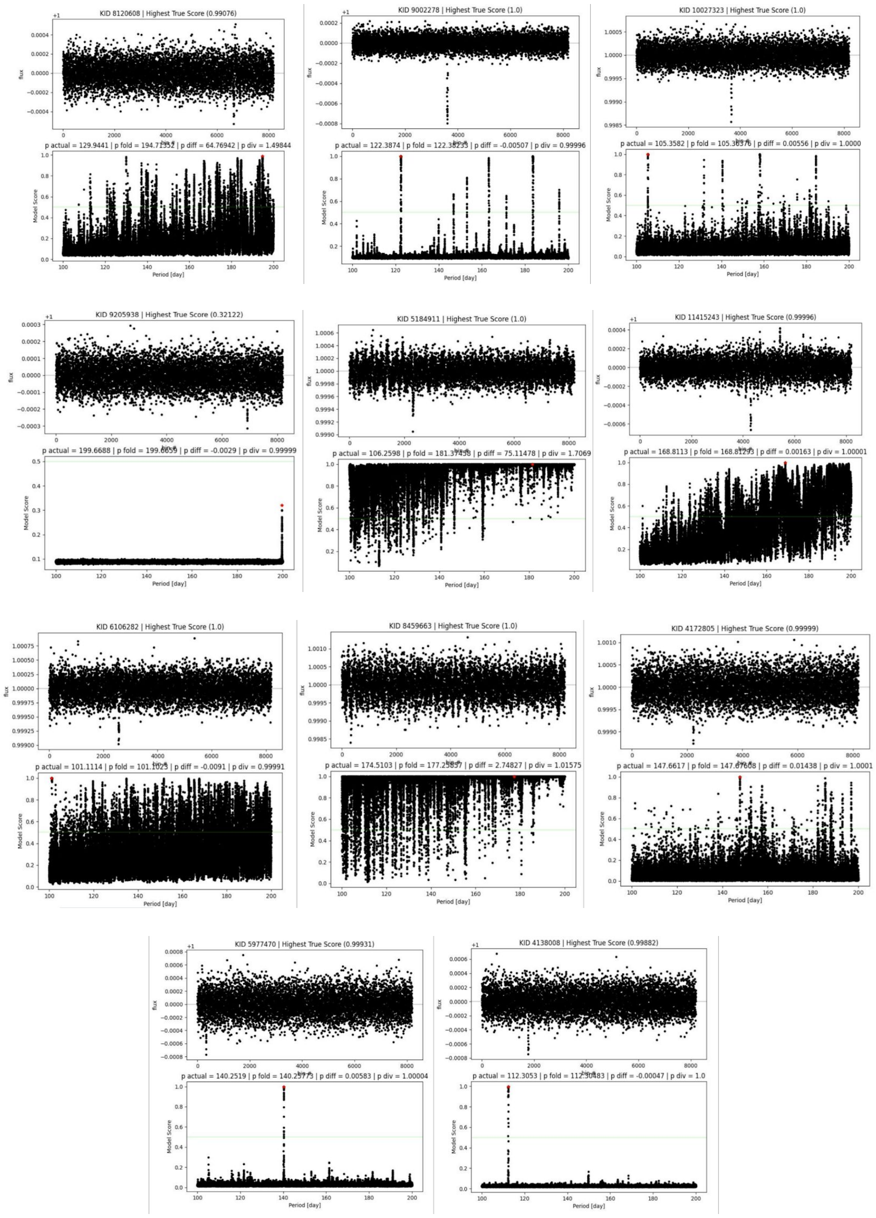


Figure 13: The neural network model's scores and results for each of the 11 HSPs in *Table 1* below. For each target, the top plot shows the folded lightcurve at the period where the model returned the maximum confidence score. The bottom plot displays the model's resulting scores across all 44,000 periods at which the lightcurve was folded. The location of the highest confidence score is marked with a red star in each plot.

*Table 1. Model Results for Recovering Known HSPs in the 100-200 Day Period Range*

Data marked with (*) indicate targets with multiple similarly-significant peaks in the model's scores. For these targets, the first value in the "Model Recovered Period" is the period originally identified by the model, and the second value is the period of the peak recovered after postprocessing that most closely matches the known period. This detected period and its corresponding score from the model are **bolded**.

| KIC | KOI | Known Period (days) | Model Recovered Period (days)* | Model Highest Score (out of 1.0)* |
|---|---|---|---|---|
| 8120608 | 571.05 | 129.9441 [1] | 194.7135*<br>**129.9438** | 0.99076*<br>**0.9826** |
| 9002278 | 701.03 | 122.3874 [4] | 122.3823 | 1.0 |
| 10027323 | 1596.02 | 105.3582 [4] | 105.3638 | 1.0 |
| 9205938 | 2162.02 | 199.6688 [3] | 199.6659 | 0.32122 (single peak) |
| 5184911 | 2719.02 | 106.2598 [3] | 181.3746*<br>**106.2592** | 1.0*<br>**0.9799** |
| 11415243 | 4036.01 | 168.8113 [2] | 168.8129 | 0.99996 |
| 6106282 | 4087.01 | 101.1114 [1] | 101.1023 | 1.0 |
| 8459663 | 4356.01 | 174.5103 [2] | 177.2586*<br>**174.5108** | 1.0*<br>**0.9999** |
| 4172805 | 4427.01 | 147.6617 [3] | 147.6761 | 0.99999 |
| 5977470 | 4550.01 | 140.2519 [2] | 140.2577 | 0.99931 |
| 4138008 | 4742.01 | 112.3053 [1] | 112.3048 | 0.99882 |

*Sources for planet periods: 1. Torres et al. 2015 | 2. Torres et al. 2017 | 3. Thompson et al. 2018 | 4. Gajdoš et al. 2019

## 7. New Habitable Small Planet Candidates

As the neural network model demonstrated that it was capable of recovering the transits of 11 confirmed small planets in the 100- to 200-day period range, searching for HSPs in data from the Kepler catalog could finally begin. To reduce the likelihood of recovering false positives, only targets with confirmed planets -- marked by a disposition of "CONFIRMED" in the Kepler catalog -- were processed. The lightcurves of 4,849 Kepler targets were fitted and normalized, and all known transits in the data were removed so the "leftover" data could be searched for planets. Of these, 148 lightcurves have been processed with the windowed fastfolding method and fed into the neural network model.

Out of the resulting model scores and identified peaks, a few targets stood out as being candidates for potential transits, which were marked for further investigation. Closer examination is required because if other periodic events, whether from planet transits or a rotationally variable star, are present in the lightcurve, it is possible that their harmonics might appear in the outputs from fastfolding and be identified by the neural network, confusing them for a real new candidate. The presence of multiple signals in a single window, or the model's scores peaking at periods corresponding to the events are both signs of possible harmonics. However, if the period identified by the model is not a multiple of any other known period in the system, it would be unlikely that the observed event is the product of a harmonic; the probability of it originating from a previously unknown transiting planet is higher. By checking each identified target of interest for other periodic events with the NASA Exoplanet Archive and the SIMBAD Astronomical Database, the list of targets with potential new HSP candidates was narrowed down to the following 6 as shown in Figure 14: KIC6276477, KIC6721123, KIC6863998, KIC10854555, KIC11812062, and KIC12644822.

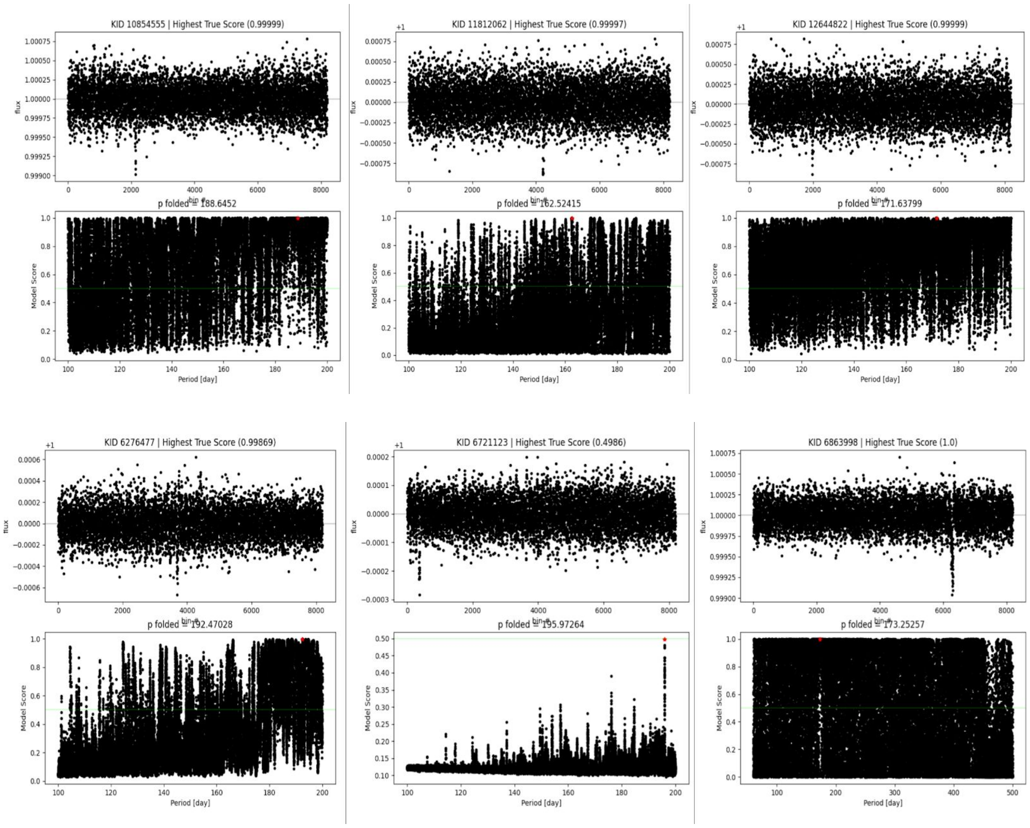


Figure 14: The 6 new potential HSP candidates recovered with the neural network model. For each target, the period at which the model scored the highest was the estimated period of the candidate; the top plot for each target shows the corresponding fastfolded lightcurve where the candidate event was identified. The bottom plot for each target shows the model's confidence scores at all 44,000 attempted periods. The location of the maximum score is marked by a red star.

To find out more about these HSP candidates, the properties of the detected signals need to be determined. First, a model is created for the potential transits with estimated parameters. Then, an LSQ solve is performed with the provided parameters to evaluate whether the initial estimates need adjustment. If the resulting transit fit appears reasonable, then the Markov chain Monte Carlo method (MCMC) was applied to solve for the period in days, the time of the reference transit, and quadratic limb darkening coefficients $b,\ u_0$, and $u_1$ , employing the parametrization introduced by Kipping, 2013. After calculating these approximate parameters, the potential transit can finally be fitted.

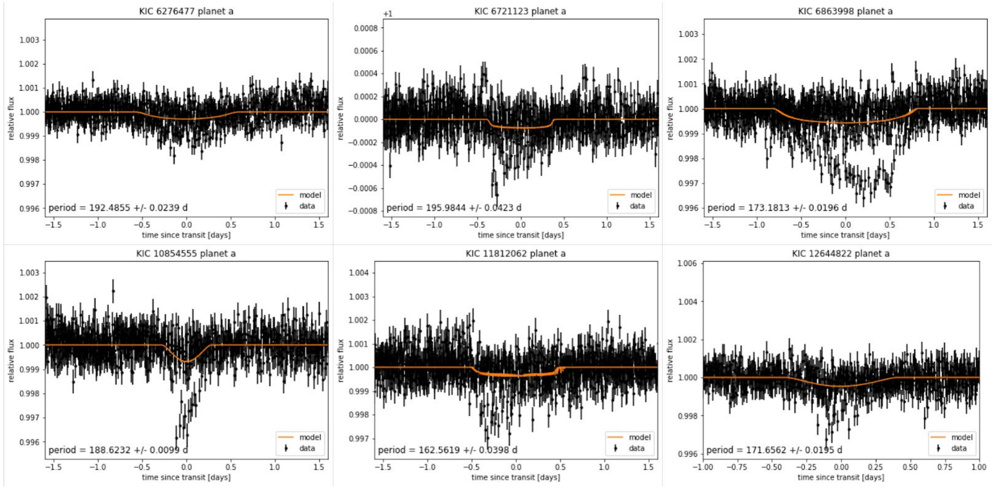


Figure 15: Fits for the 6 HSP candidates' transits, shown in orange, were generated with properties solved with the MCMC method.

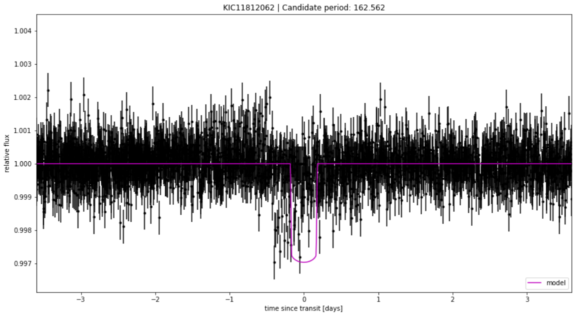


Figure 16: The initial step of the process to solve for the parameters of the potential transit in KIC11812062 is to determine "initial guess" values as a starting point for the MCMC solver. The magenta line marks the signal as modeled by the "initial guess" parameters for limb darkening values, period, and $t_0$ .

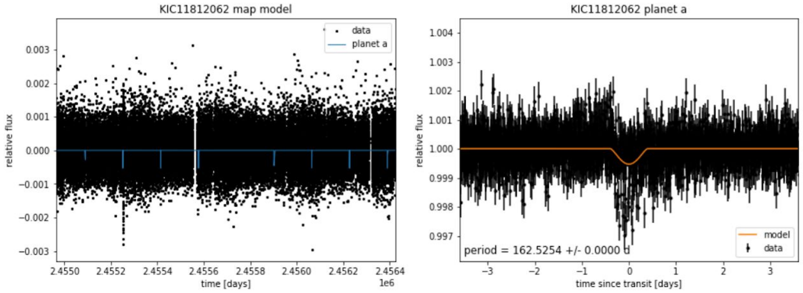


Figure 17: After creating a model for the possible transit in KIC11812062, an initial LSQ solve is performed. If the solve is successful with the previously given parameters -- a period is present in the left plot and  the fit for the transit in the right plot looks reasonable, the next step of solving with the MCMC method can be run. If not, then the initial "guess" values  need to be adjusted and the solve attempted again.

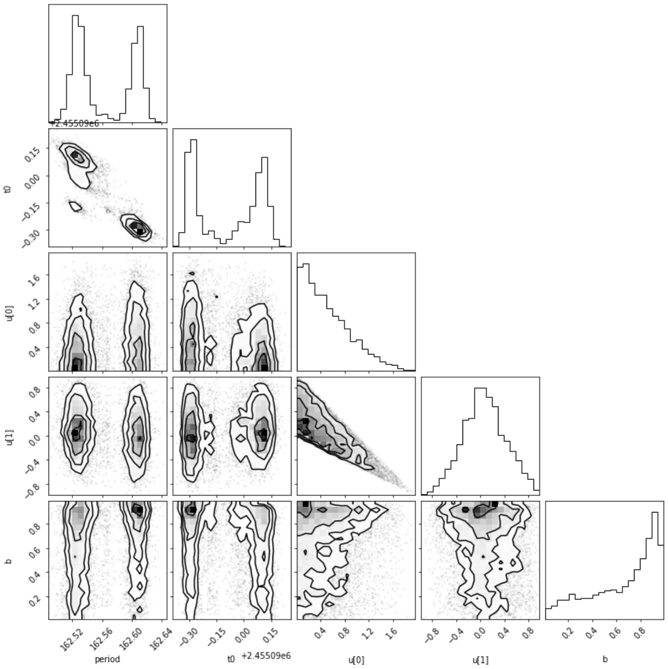


Figure 18: The probability distributions for the MCMC solver's attempt to calculate the parameters for the potential transit in KIC11812062. From left to right, the plots show the probability distributions and likelihood correlations for orbital period in days, $t_0$ , and limb darkening parameters $u_{0,}u_1$, and *b*.

### *Table 2. Transit Parameter Summary for 6 New HSP Candidates*

| **Target KIC** | **Period (days)** | $t_0$ **(bkjd)** | ***b*** | $u_o$ | $u_1$ | **Detection significance** |
|---|---|---|---|---|---|---|
| 6276477 | 192.482 ± 0.021 | 2455142.497 ± 0.040 | 0.838 ± 0.241 | 1.014 ± 0.492 | − 0.212 ± 0.440 | +0.922**σ** |
| 6721123 | 195.984 ± 0.042 | 2455061.507 ± 0.053 | 0.771 ± 0.224 | 0.534 ± 0.417 | 0.033 ± 0.363 | +19.2**σ** |
| 6863998 | 173.181 ± 0.020 | 2455032.745 ± 0.042 | 0.732 ± 0.257 | 0.777 ± 0.485 | 0.016 ± 0.460 | +0.833**σ** |
| 10854555 | 188.623 ± 0.010 | 2455172.251 ± 0.036 | 1.057 ± 0.058 | 0.848 ± 0.537 | − 0.113 ± 0.446 | +0.628**σ** |
| 11812062 | 162.562 ± 0.040 | 2455089.910 ± 0.181 | 0.664 ± 0.270 | 0.510 ± 0.393 | 0.063 ± 0.353 | +0.870**σ** |
| 12644822 | 171.656 ± 0.020 | 2455068.516 ± 0.092 | 0.989 ± 0.128 | 0.818 ± 0.528 | − 0.075 ± 0.454 | +0.711**σ** |

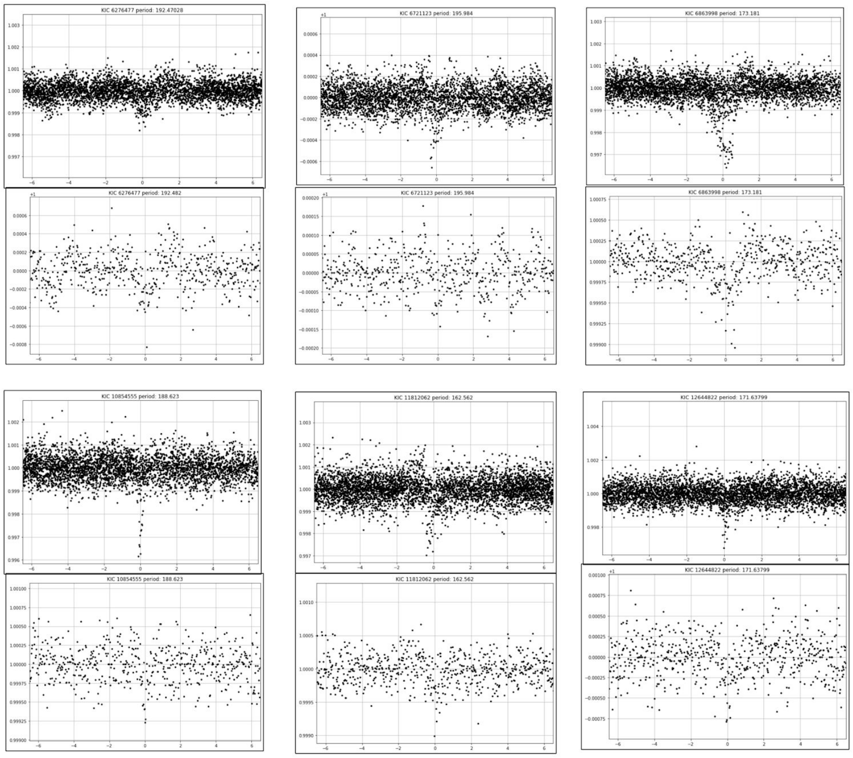


Figure 19: The candidates' transits, manually folded and binned to verify the presence of a feature. On the top row, from left to right: KIC6276477, KIC6721123, and KIC6863998; the bottom row shows, from left to right: KIC10854555, KIC11812062, and KIC12644822. The top plot for each target shows a "global" view of the folded potential transit, and the bottom plot shows a "local" view. A bin size of 8,192 was used for each fold, staying consistent with the representation of the data in the fastfolding outputs.

## 8. Future Work

### *8.1 Further Study of Candidates*

Further analysis of the six detected HSP candidates is necessary to demonstrate their validity and determine more of their properties. Closely studying the shapes of the signals with a tool like the BLENDER technique (Torres et al. 2017), which attempts to determine the probable factors causing a candidate transit signal's shape by modeling the effects of different eclipsing or transiting objects, ellipsoidal variabilities, and variables like orbital eccentricity or distance, can help validate these candidates and determine the probability of the signals originating from real planets rather than from another source. The preliminary evaluation of the candidates provided estimates for transit properties like period, initial reference time, and limb darkening parameters, but more information recovered about the targets may lead to adjustments or recalculations of these values. Combined with other follow-up observations of the signals and stars, if any of these signals are found to be very likely new planets, the properties of the star and surrounding environment could be analyzed as well, to better evaluate the habitability of the planet candidate.

### *8.2 Improved Filtering*

A future improvement that might lead to more recovered positive signals is to better account for other fixed-period events in a stellar system. For example, many targets in the Kepler catalog are rotationally variable stars. For these stars, their observed brightness varies significantly as they rotate. Whether this is due to an ellipsoidal shape or the prominent presence of spots and patches on the star, this can result in dips in observed brightness similar to those originating from a transiting planet. Although this has not currently been implemented in this process, one possible solution for removing this periodic noise is to use a Fourier series with terms corresponding to the frequencies for the known fixed periods to model the variations and filter them out during preprocessing.

### *8.3 Continuing the Search*

Currently, this search has only included a fraction of all possible planet hosts and types of planets. As the methodology presented in this paper has only been applied to model and search for small planets with periods between 100 days and 200 days so far, there is a lot of future work to be done in extending the process to include a wider range of orbital periods. For example, HSPs and HSP candidates with periods ranging from under 20 days (Gajdoš et al. 2019) to nearly 400 days (Jenkins et al. 2015) have been discovered; in fact, if the optimistic habitable zones are taken into account, then periods of over 800 and 1000 days may even be possible for planets orbiting K- and F-type stars respectively. The specific properties of planets with those larger orbits would require further study, but with modifications, a similar process and program might be usable in detecting and producing many more such candidates than are currently known. Additionally, beyond expanding the range of periods to be searched, another area of future work is to expand beyond the Kepler catalog to find targets, as this approach with normalizing, windowed fastfolding, and using a neural network model could be applied to data from TESS as well.

## 9. Conclusion

In this paper, a methodology for improving the efficiency of processing and analysis in detecting habitable small planets in Kepler data is introduced. At its core is a convolutional neural network model designed and trained to recover the transits of potential HSP candidates from Kepler lightcurve data. Improvements were made to the process of obtaining input data with the development of the new windowed fastfolding method, which can prepare lightcurves for input into the neural network much more efficiently than manual folding using periods solved with BLS, while also remaining more sensitive to weaker transit signals than the traditional method with BLS. Although the very limited number of known HSPs presented a challenge to creating a sufficiently large training dataset for the neural network, this was remedied by the creation of a simple simulation for transit signals, which generated hundreds of thousands of realistic positive samples used for training. Though there is still room for improvement regarding the model's validation accuracy, it was able to reach a training accuracy of

99.04 percent after training for 100 epochs, and proved its capability by successfully recovering from fastfolded lightcurve data the transits of 11 known HSPs with periods in the range of 100 days to 200 days. A search was then conducted to detect new HSP candidates in the 100- to 200-day period range, examining targets in the Kepler catalog with already-known and confirmed planets. Postprocessing the neural network model's output scores allowed for a large number of interesting signals to be presented for closer inspection, with the intent of picking out signals that may have been previously overlooked. After these identified signals were filtered to remove those resulting from harmonics, strong signals which did not correspond to any known features were found in the six targets KIC6276477, KIC6721123, KIC6863998, KIC10854555, KIC11812062, and KIC12644822. After some examination, these six signals were identified as new potential HSP candidates. Though only a small portion of Kepler targets and a subset of possible HSPs were studied, the neural network model and the improved processing and analysis methodology have demonstrated promising potential in the search for HSPs. The model and approach can be improved in the future to expand the search to even longer-period planets, covering wider sections of the habitable zones of FGK main-sequence stars. The six new HSP candidates detected by the model also require further study for confirmation and analysis. As more HSP candidates are discovered, analyzed, and confirmed, and the population of known HSPs grows, a door to learning about the development, evolution, and properties of habitable environments is opened, which could help shed light on Earth's place in the universe and the possibility of life occurring elsewhere. How common are habitable planets? Is Earth a typical example of this population, or is it an outlier? Though these questions and more like them are currently unanswered, every new discovery is a step closer.

# References and Acknowledgments

Gajdoš, P., Vaňko, M., & Parimucha, Š, 2019. “Transit Timing Variations and Linear Ephemerides of Confirmed Kepler Transiting Exoplanets”. *Research in Astronomy and Astrophysics*, 19(3), 041. https://doi.org/10.1088/1674-4527/19/3/41

Jenkins, J. M., Twicken, J. D., Batalha, N. M., et al. 2015, “Discovery and Validation of Kepler-452b: A 1.6R⊕ Super Earth Exoplanet in the Habitable Zone of a G2 Star”. *The Astronomical Journal,* 150(2), 56. https://doi.org/10.1088/0004-6256/150/2/56

Kipping, D. M., 2013. “Efficient, Uninformative Sampling of Limb Darkening Coefficients for Two-Parameter Laws”. *Monthly Notices of the Royal Astronomical Society*, 435(3), 2152–2160. https://doi.org/10.1093/mnras/stt1435

Kopparapu, R. K., Ramirez, R., Kasting, et al. 2013, “Habitable Zones Around Main-Sequence Stars: New Estimates”. *The Astrophysical Journal*, 765(2), 131. https://doi.org/10.1088/0004-637x/765/2/131

Kovács, G., Zucker, S., & Mazeh, T, 2002. “A Box-Fitting Algorithm in the Search for Periodic Transits”. *Astronomy & Astrophysics*, 391(1), 369–377. https://doi.org/10.1051/0004-6361:20020802

Thompson, S. E., Coughlin, J. L., Hoffman, K., et al. 2018, “Planetary Candidates Observed by Kepler . VIII: A Fully Automated Catalog with Measured Completeness and Reliability Based on Data Release 25”. *The Astrophysical Journal Supplement Series*, 235(2), 38. https://doi.org/10.3847/1538-4365/aab4f9

Torres, G., Kipping, D. M., Fressin, F., et al. 2015, “Validation of 12 Small Kepler Transiting Planets in the Habitable Zone”. *The Astrophysical Journal*, 800(2), 99. https://doi.org/10.1088/0004-637x/800/2/99

Torres, G., Kane, S. R., Rowe, J. F., Batalha, et al. 2017, “Validation of Small Kepler Transiting Planet Candidates in or near the Habitable Zone”. *The Astronomical Journal*, 154(6), 264. https://doi.org/10.3847/1538-3881/aa984b

This research made use of the exoplanet package and its dependencies PyMC3 for the inference engine and modeling framework, AstroPy for units and constants, Kipping 2013 for the reparameterization of the limb darkening parameters for a quadratic law, and Luger et al. (2018) for the light curve calculation.

This research made use of Lightkurve, a Python package for Kepler and TESS data analysis (Lightkurve Collaboration, 2018), as well as its dependencies AstroPy and AstroQuery.

This research has made use of the NASA Exoplanet Archive, which is operated by the California Institute of Technology, under contract with the National Aeronautics and Space Administration under the Exoplanet Exploration Program.

This research has made use of the SIMBAD database, operated at CDS, Strasbourg, France.
*2000,A&AS,143,9*, "The SIMBAD astronomical database", Wenger et al.