

Flicker-Based Estimation of Stellar Surface Gravity Using Transiting Exoplanet Survey Satellite (TESS) Light Curves

DONOVAN QUIMBY^{1,2}

¹*Georgia Institute of Technology*

²*Institute for Data Intensive Research in Astrophysics and Cosmology*

ABSTRACT

Stellar surface gravity is a fundamental star property that is essential to understand in astronomy and astrophysics. However, traditional methods of modeling or directly measuring stellar surface gravity are expensive, have low accuracy, or are only practical for a small subset of observed stars. Previous research has shown that relatively accurate estimations of surface gravity are possible using the stochastic brightness variations, called flicker, in time-series images of stars. However, this method has only been demonstrated on a small dataset of relatively high signal-to-noise ratio (SNR) images from the NASA Kepler mission. This work aims to build on the previous work by applying flicker-based surface gravity modeling theory and techniques to the NASA Transiting Exoplanet Survey Satellite (TESS) mission dataset, which contains a substantially larger database of stars. Inherent differences in the architecture of the TESS mission telescope may decrease the signal-to-noise ratio of the images, and it is unclear if stellar gravity estimation via flicker is possible with this data. We developed a python-based data analysis tool to download, preprocess, explore, calculate flicker, and model surface gravity using TESS light curve data. We model the relationship between flicker and surface gravity using polynomial, LOWESS, and support vector regression models. This work finds that the TESS data has significantly lower SNR than the Kepler mission data, which causes the irreducible shot noise to overwhelm the flicker signal. The low SNR poses a significant hurdle to using flicker-based models to estimate stellar surface gravity with the TESS dataset. We demonstrate that the excessive noise in the TESS data leads to regions of sparse data with substantial outliers, which causes overfitting of the gravity models leading to results that are not physically representative of the expected behavior. We supply possible avenues of study for future work that may help overcome the SNR noted in this report and improve the flicker-based modeling accuracy for TESS data.

1. INTRODUCTION

Studying stars, stellar populations, and the physical properties that govern their formation, behavior, and death is essential to advancing fields such as astronomy and astrophysics. Stellar surface gravity is one of a star's fundamental properties, yet understanding of this elementary characteristic is currently limited. Typical methods used to measure a star's gravity, such as spectroscopy and photometrics, result in estimation accuracies of 25-50% and 90-150% respectively (Valenti & Fischer

2005; Ghezzi et al. 2010). Other methods, such as asteroseismology, offer significantly more accurate measurements, on the order of 2%, but are only available for use on a minimal subset of stars (Chaplin et al. 2011; Huber et al. 2011; Stello et al. 2013). Bastien et al. (Bastien et al. 2013) demonstrated that variations in the brightness, referred to as "flicker," of sequential images of stars can be used to estimate stellar surface gravity. Further refinement of this technique showed that this methodology could estimate surface gravity at an accuracy of approximately 1% (Bastien et al. 2016). However, the flicker-based methodology's development and application have been limited to high-quality data obtained during the NASA Kepler space mission, which contains image sequences for only a relatively small number of stars, $\approx 150,000$.

The NASA Transiting Exoplanet Survey Satellite (TESS) is a space-based telescope with a primary mission of searching for extra-solar planets via asteroseismic methodologies using time-sequential images of stars similar to those of the Kepler mission (Ricker et al. 2014). The TESS mission was launched on April 18, 2018, and its primary mission is scheduled to be completed in September 2022. A tool and model that can quickly and accurately estimate surface gravity based on the data collected by the TESS mission will be highly beneficial for advancing research in the astrophysics and extra-solar planet communities. However, fundamental differences in the telescope design used for the Kepler and TESS mission may prevent the use of TESS data for flicker-based modeling of stellar surface gravity. The work presented in this report aims to develop a tool that can efficiently process the TESS data and assess the validity and accuracy of modeling stellar surface gravity using flicker measurements.

Although studied extensively, the physics and properties governing the structure and evolution of stars are only understood on a general level. Methods such as photometry, spectrometry, and interferometry have been used to infer some stellar physical properties with poor precision (Aerts et al. 2010). More recently, the field of asteroseismology has made significant advancements in the precision and understanding of stellar properties by analyzing the oscillations of stars via sequential images collected from satellite telescopes. For this work, we use methods developed in the broader field of asteroseismology to analyze the stochastic variation in brightness of a subset of images from the TESS mission. Furthermore, we assess if it is possible to estimate the surface gravity of stars in the TESS catalog using flicker-based models.

The Transiting Exoplanet Survey Satellite has an array of four identical cameras that take a continuous stream of images of a $24^\circ \times 96^\circ$ sector of the sky as it orbits the earth. Each camera is identical and is composed of 4 2000x2000 CCDs which are sensitive to light in the blue to near-infrared ($600 - 1000\text{nm}$) spectrum (Isaac 2022). Each predetermined sector is observed for approximately 27 days, after which the satellite repositions to observe a new sector. Certain sections of the sky overlap for multiple sectors, resulting in a section of the sky known as the continuous viewing zone (CVZ), as shown in Figure 1. The current work focuses on stars residing in the CVZ because the longer observation periods result in significantly more data for each star.

All four TESS cameras take continuous streams of sector images with 2-second exposures. The 2-second exposures are processed into two separate data products, Full Frame Images (FFI) and Target Pixel Files (TPF). This work focuses on the TPF, but we briefly discuss FFIs for didactic purposes. An FFI is the co-adds, or pixel by pixel summation, of 30 minutes of continuous 2-second exposures of the entire sector. An example FFI image is shown in the example in Figure 2a. TPFs are target-specific images that mask out all data not specific to a target from the larger FFI. FFIs

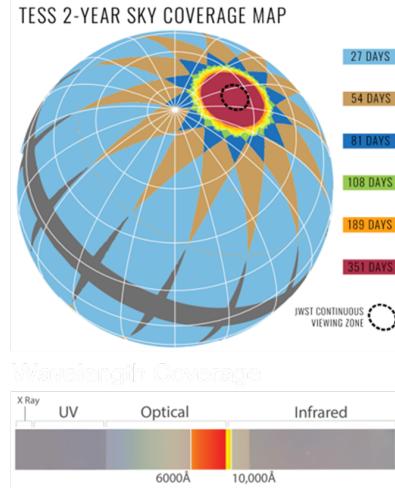


Figure 1. Map of observations and wavelength coverage for primary TESS mission (STS 2022).

are 2-minute co-adds of a predefined mask that isolates each target of interest. The TPF data is made available as a time-series sequence of images in a Flexible Image Transport System (FITS) file format, which includes both the image files and related meta-data. An example TPF file image extracted from the larger FFI file is shown in Figure 2b.

For the analysis presented in this work, we will ultimately analyze light curves created from the TPIs. Light curves are time-series data representing a target's brightness. Light curves are created by summing the flux value (intensity) of all pixels in each individual time-series image of the TPF files. The summation results in a single flux value for the target at each time in the time-series. This work considers a 2-minute interval TPF, and therefore the time interval for the light curve data is 2 minutes. Furthermore, the target stars we consider are imaged over multiple sequential sectors, increasing the time-series data length. Figures 2d and 2c show examples of light curves for single and multiple sectors, respectively.

It has been shown that the Fourier transform of the power spectrum of granulation, the alternating patterns of bright and dark cells on the surface of a star due to the convection of hot gases, correlates with stellar surface gravity (Mathur et al. 2011; Kjeldsen & Bedding 2011). However, precise observation of surface granulation is difficult and limited to a tiny subset of close stars. It has also been shown that relatively high-frequency brightness variations on the order of hours are partly caused by granulation (Brown et al. 1991). Bastien et al. found that the standard deviation of flux driven by granulation on an 8-hour time scale, referred to as "flicker" (F_8), can accurately be used to estimate \log_{10} stellar surface gravity using light curves from the Kepler mission (Bastien et al. 2013, 2016) with a precision of 0.1 – 0.2 dex, where 1 dex is a non-dimensional order of magnitude.

This report expands the flicker technique to TESS mission light curves to create a surface gravity model for TESS targets. However, it is not apparent that this will be possible for several reasons. For one, the size of the sensors in the TESS mission is significantly smaller than those used in the Kepler mission, reducing the flux magnitude and significantly decreasing the signal-to-noise ratio. Furthermore, a "gold standard" of surface gravity labels does not currently exist for the stars studied in this work. Estimations of surface gravity used for modeling purposes are based on models with low levels of precision. Therefore, this work aims to create a data processing tool that can efficiently

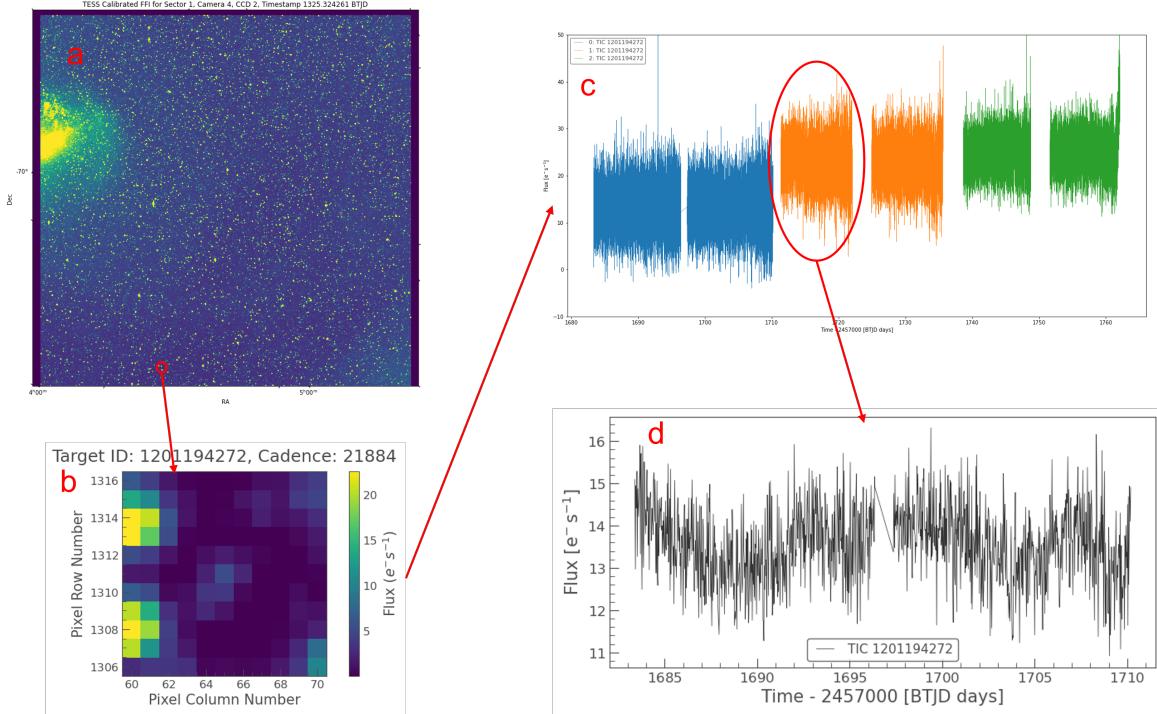


Figure 2. Examples of a) FFI, b) TPF, c) light curve plot of multiple sectors, and d) single light curves are shown for reference. Outliers in the single light curve plot and time gaps in the data between multiple sectors and within the single sector plot are noted.

download, process, and explore the TESS data and model and assess the viability of a surface gravity model based on the resulting flicker determined by the tool. The remainder of this report discusses the development of the data processing tool, the results of modeling surface gravity with the calculated flicker, and the conclusion and future work that can address issues discovered in this report.

2. DATA ANALYSIS

This section describes the data acquisition, processing, and exploration tool development. All analysis for this work is performed using Python 3.8 (Van Rossum & Drake 2009). Due to the possible deployment of this tool on large datasets, over 100m stars, the computational efficiency of all steps were considered. All data used in this project is accessed via the Mikulski Archive for Space Telescopes (MAST) database (Marston et al. 2018). The data processing pipeline consists of 8 steps: data acquisition, normalization, imputation of missing data, low-frequency cyclic trend removal, outlier removal, smoothing, shot noise removal, and the final calculation of the target stars flicker.

2.1. Data Acquisition

The first step of developing the data analysis tool was identifying stars within the TESS CVZ. Identifying the stars was accomplished by analyzing the metadata of all stars observed in the first 27 sectors of TESS data. The full dataset of stars consists of approximately 493k stars, of which 227k are unique. A heatmap of the location of all unique stars in the sky is provided in Figure 3a below, where the count is determined by summing the number of stars in bins created by dividing the Right Ascension and Declination into 100 bins. Areas of high concentration correspond to locations

near the Earth's poles and cover the continuous viewing zones as shown in Figure 1. We isolate stars of interest by selecting ones with data for 10-13 sectors, with the lower limit identifying stars within, or close to being within, the continuous viewing zone. The upper limit filter stars with source data anomalies because stars in the CVZ would not have more than 13 sectors due to line of sight limitations. We identify a total of 11,832 stars that are used for the remainder of this work. A final check is conducted by plotting the location of the filtered stars of interest and assuring that they reside within the expected CVZ region, Figure 3b.

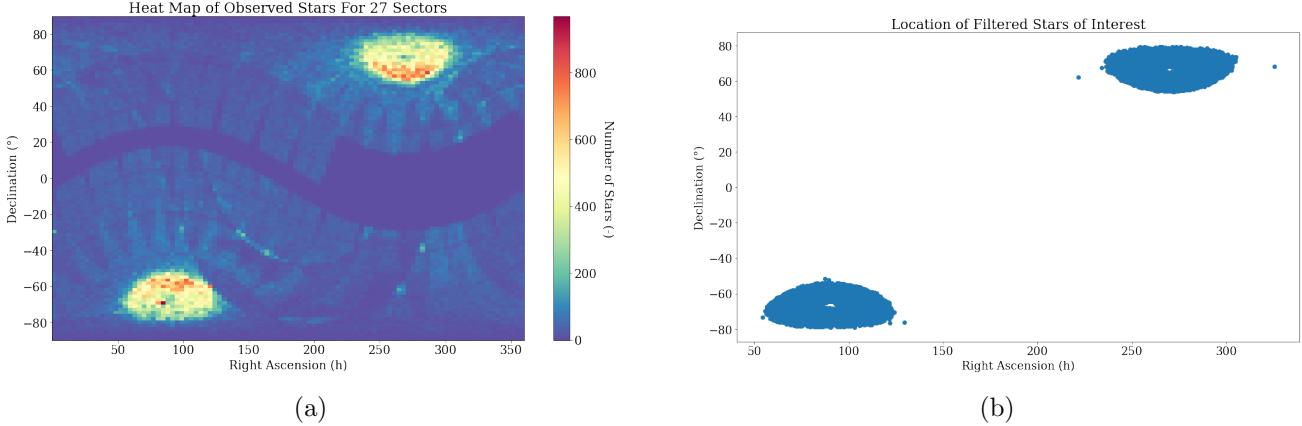


Figure 3. a) Heat map of the number of stars for the first 27 sectors for the TESS mission. b) Filtered stars of interest which line up with the expected CVZ location.

After identifying the stars of interest, the corresponding light curve data must be downloaded and stored for each star. Data is accessed through an API for TESS data at the MAST archive. We download data from the Science Processing Operations Center (SPOC) pipeline (Caldwell et al. 2020) for this work, which is the official raw data processing pipeline for TESS light curves. SPOC processing includes selecting and masking TPIs from FFIs, raw data calibration based on known instrumentation corrections, metadata assignment, and identification and tagging of known anomaly events described in (Twicken et al. 2020). 120s (2-minute) interval light curves for target stars are downloaded and stored locally. All extraneous data except for flux as a function of time, measured in units of $\frac{e^-}{s}$, is stripped to reduce local storage size and memory usage. Searching and downloading the data for the $\sim 12,000$ stars through the API, as discussed above, took approximately 40 hours. We identified database search functionality and traffic on the database server as significant contributors to the download time required. In future work, other API and archive access strategies will be investigated to reduce the time for bulk data analysis.

For the gravity modeling portion of this report, we need labeled data that corresponds to the TESS data of interest because no direct measures or estimates of gravity are available with the TESS data. We use gravity estimations from cross-matched targets in the Gaia release 3 databases, also located at the MAST archives (Gaia & Bono 2016; Brown et al. 2021). The Gaia release 3 data is a new data set released on June 13, 2022. Further details are discussed in section 3.

2.2. Data Normalization

The estimation of surface gravity by brightness variation presented in this paper is essentially a point estimation of the standard deviation (σ) of the light curve on time scales of hours, more specifically,

8 hours, based on previous work. The first step in conducting this analysis is to normalize the flux data and convert it into units of parts per thousand (ppt) for comparison to data, methods, and results of other published work. Normalization is accomplished by dividing a light curve by its median value, calculated while ignoring missing values, and multiplying by 1000. It should be noted that all analysis presented in this work is done on a per sector, 27-day basis unless otherwise noted. Therefore any reference to a light curve is specific to the data from a particular target star for a particular sector. This protects the data time structure for future work beyond this project’s scope, allowing for analysis on sector length time-scale.

2.3. Data Imputation

After normalization, we address missing data in the light curves. Data obtained from the TESS library contains numerous occurrences of missing data. Two practical methods of dealing with missing data are considered: removing missing data entries and imputing data. Although point estimations of various moments of the light curves distributions ultimately used in this analysis do not explicitly rely on the encoded time aspect of the data, pre-processing steps such as the low-order filtering discussed below do, and therefore simply eliminating missing data is not an option. We, therefore, employ imputation techniques for estimating and imputing missing data.

Two different imputation methods were implemented and evaluated using quantitative and qualitative metrics. The first method was imputation utilizing linear interpolation. Linear interpolation proved an effective means of filling missing data between small gaps. However, the stochastic variation of the data meant that interpolation across more significant gaps in the data encoded unrealistic linear assumptions into the data. Additionally, implementations of this imputation method did not have satisfactory means to deal with missing edge data leading to the choice of using either the closest observed value or unbounded extrapolation. Using the closest value often adds a significant number of data points at very high or low magnitudes, which would skew a standard deviation measurement. The unbounded extrapolation often leads to extreme estimations well beyond realistic flux values, as demonstrated in Figure 4a.

The second method investigated was to impute the data using the median value. This method makes no assumptions about the time-dependent behavior of the data and leads to reasonable estimations for edge data, Figure 4b. Also, imputation by the median method was roughly 32% faster than the linear interpolation method. For the above reasons, the median imputation method was chosen for implementation in the tool. We recognize that this method will artificially lower the SD, but initial investigations show the reduction is generally negligible. Other methods may also be considered in future work.

2.4. Low Frequency Trend Removal

Modeling stellar surface gravity by flicker relies on the physics-based assumption that variation in the light curve is granulation based and therefore encoded in the time period of hours. However, techniques described in section 2.6 intended to isolate hour-order variation often fail to remove low-frequency trends that occur over days or weeks. To address this, a low-order model is fit to the data and subtracted to remove the low-frequency trends. Two different models are evaluated and discussed below.

A simple moving average function is fit to the data using various user-defined window sizes that equate to time frames of days. Likewise, a natural cubic spline smoothing function is fit with the

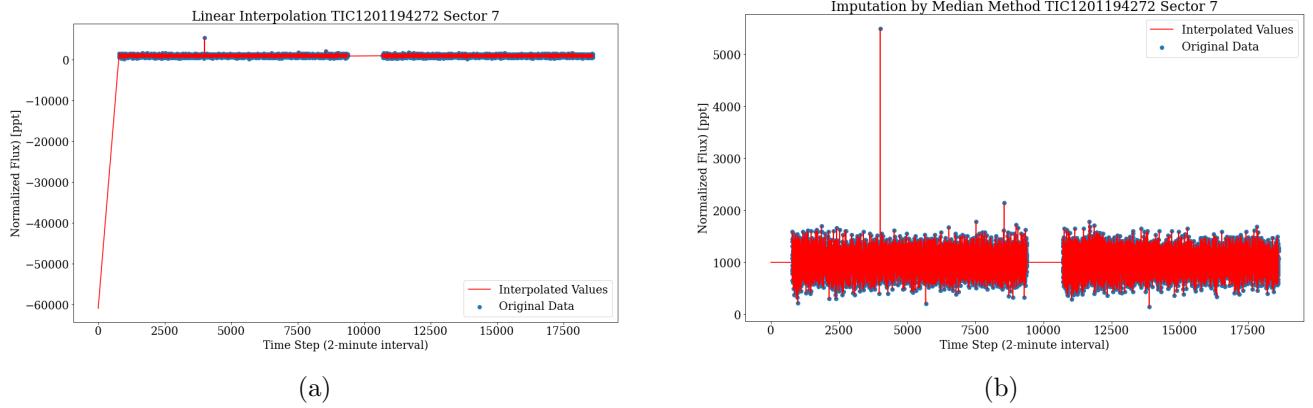


Figure 4. Comparison of the imputed data in red plotted over the original data for a) the linear interpolation method showing unbounded extrapolation and b) the median method for the same data. Note that imputed data is only used in locations of missing data.

number of knots calculated to be equivalent to the time frames tested in the moving average function. For the moving average function, endpoints are treated using a mirroring method, which extends the input array beyond the boundary point by reflecting about the center of the terminal endpoint. The natural cubic spline handles endpoints by forcing the second derivative of the smooth fit to be 0 at the endpoints.

The general procedure for tuning the fit and evaluating either of the functions would typically be through training, testing, and evaluation sets in conjunction with cross-validation for tuning. However, this generalized methodology is not viable because we are using this as a filtering function rather than in the more general regression case. We instead evaluate the functions based on qualitative observations of specific stars and sectors with known low-frequency oscillations like the example shown in Figure 5, which displays a cyclic pattern repeating approximately every 15 days.

From a qualitative evaluation perspective, both functions do the intended job of capturing and removing the low-frequency trends. However, for a given equivalent time window, the natural cubic spline fit tended to give a slightly smoother fit than the moving average. On the other hand, the spline fit took approximately 3x longer to run. The moving average method was chosen as the primary low-frequency removal method due to its computational efficiency and the expectation that this tool will be used on large datasets processing bulk data. However, the natural cubic spline is left in the tool as an optional method for future use. For the remainder of this work, we implement the moving average method with a window of 72 hours.

2.5. Outlier Removal

The feature of interest in this analysis is the standard deviation, σ , of a light curve defined as flicker, F_n , where n is the maximum time-scale period of interest on the scale of hours. The σ estimation of F_n is not robust to outliers in the data, which may be caused by events unrelated to the granulation-driven physics correlated to surface gravity. Although many methods are available to identify and remove outliers, this work implements a sigma-clipping method. Sigma-clipping is an iterative process by which all data points with values greater or less than a specified number of standard deviations from a center value are identified and removed from the dataset. The process is repeated over the new data until convergence, or a user-defined number of iterations is reached. In

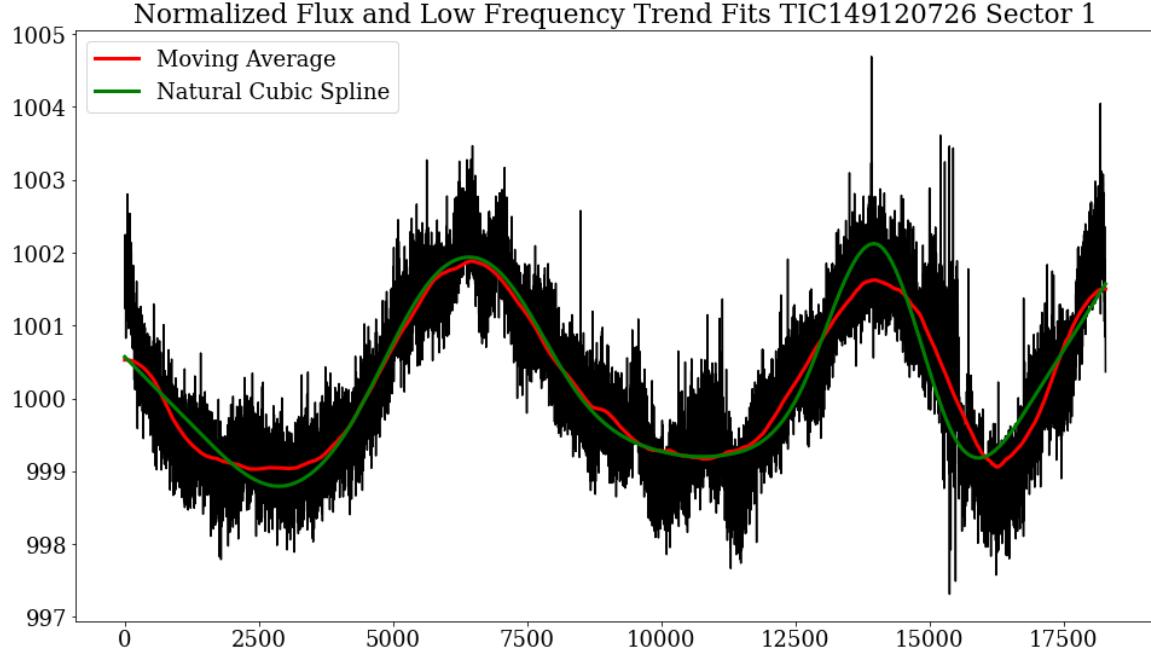


Figure 5. Example light curve displaying a low-frequency cyclic trend repeating approximately every 15 days. A moving average fit with a window of 72 hours and a natural cubic spline fit with the number of knots creating an effective window of 72 hours are shown in red and green, respectively.

this work, we use the median as the center function and let the algorithm run to convergence. We use 2.5σ as the cutoff value with the expectation that it will remove the 1% most outlying points from the population. Figure 6 shows a typical light curve before and after sigma-clipping.

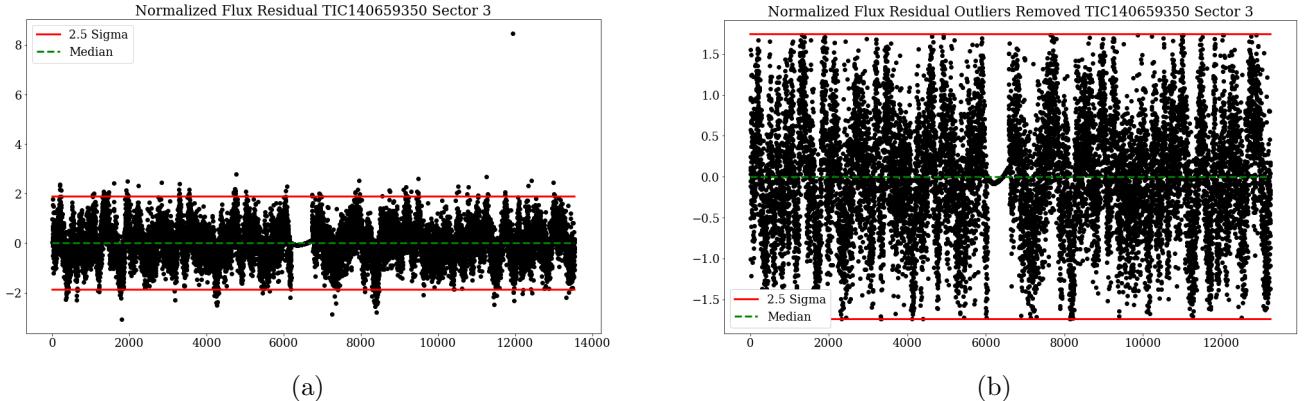


Figure 6. Example plots of sigma clipping on a light curve with outliers. a) shows the original residual from the normalized light curve with the low-frequency trend removed and the calculated median and 2.5σ values. b) shows the residual from the normalized light curve with the low-frequency trend removed and the sigma-clipping algorithm fully converges.

2.6. Smoothing

Light curve flicker is determined on a time scale of $n = 8$ hours. In order to perform the flicker calculation, we first calculate the residual of the light curve following the subtraction of an 8-hour

smoothing filter, which isolates the σ for the time scale of interest. The smoothing is accomplished by applying a 1-dimensional uniform moving average filter with a window sized to span 8 hours of data. The fit is subtracted from the light curve resulting in the 8-hour smoothed residual. Figure 7a shows an example of the smoothing process on a star that has a noticeable cyclic variation on a time scale shorter than the low-frequency filtering would remove but larger than the 8-hour period of interest. The 8-hour smooth fit removes the cyclic behavior when subtracted from the original preprocessed flux, resulting in the final flux residual used to calculate flicker, Figure 7b.

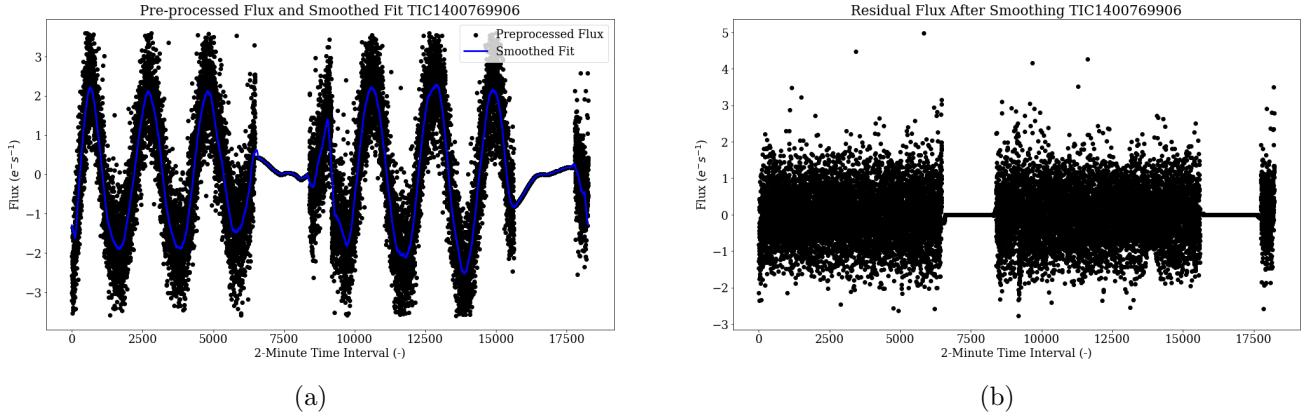


Figure 7. Example plots of 8-hr smoothing. a) shows the original flux after all previously discussed preprocessing has been applied in black, and the 8-hr smoothing curve fit to the data in blue. b) shows the final flux residual after subtracting the 8-hour smoothed fit from the preprocessed data.

2.7. Flicker Calculation

F_8 flicker is calculated by determining the standard deviation of the resulting residual flux after subtracting the 8-hour smoothing curve. For each star, we determine the standard deviation for each sector and take the median value of all sectors as the final F_8 value. A natural extension of this method is to see if higher-order moments of the residual distribution contain any other information that may help estimate a star's surface gravity. Therefore the residual skew and kurtosis are also determined in the same manner as the standard deviation and reserved for future studies.

The sector-based distribution moments are indexed to their respective sector and saved, forming a secondary time-series dataset. Future work may involve studying how these point estimates of the distribution vary over time and what additional information regarding the physical characteristics of stars can be derived from it.

2.8. Shot Noise Correction

In order to represent the physical variations in brightness due to stellar surface granulation, it is essential to remove noise from the sensors and hardware, notably shot noise. The images used to calculate the light curves in this analysis are made by CCD camera sensors that detect photons (light) emitted from a stellar source. However, stars emit photons randomly, so the photon arrival rate at the telescope CDD sensors is also random. The random arrival of photons to the CCD sensor introduces noise into the flux measurements called shot noise. The shot noise is the minimum irreducible amount of noise in the observation. Removing the contribution of shot noise from the

overall variation is crucial to getting an accurate physical representation of the granulation-driven flicker.

To remove the shot noise, we first recognize that shot noise contribution to the flicker is a function of the star's observed brightness. For all stars in the data set of interest, we cross-match luminosity values available in the MAST data repository, which is an inverse measure of brightness. We then fit a quadratic regression curve to the bottom 0.5 percentile of the calculated $\log_{10}F_8$ versus the apparent luminosity as discussed in detail in (Bastien et al. 2013, 2016).

The calculation is performed by first binning the $\log_{10}F_8$ values into luminosity bins of 0.1 G-magnitude value, where luminosity varies between \sim 7-14. We then calculate each bin's bottom 0.5 percentile and fit a third-order polynomial to these values. The third-order polynomial fit represents the observed shot noise floor. The shot noise correction to observed F_8 is performed in quadrature as shown in Equation 1. All references to flicker or F_8 for the remainder of this report refer to shot noise corrected values.

$$F_{8,corrected} = \sqrt{F_{8,observed}^2 - (10^{0.00002 - 0.005*lum + 0.086*lum - 1.55*lum})^2} \quad (1)$$

Figure 8 shows a scatterplot of the TESS data $\log_{10}F_8$ versus apparent luminosity, the empirically derived TESS shot noise floor as described above, and the equivalent Kepler empirically derived shot noise floor calculated by Bastien et al. (Bastien et al. 2016). Two critical observations of note can be obtained from the analysis of this plot: The Kepler shot noise floor is significantly lower than the floor for the TESS data, and the TESS scatter plot data is highly concentrated in the region of the floor. Both observations highlight noise inherent to the TESS data, which will likely be problematic to the modeling we wish to perform in this work which is discussed in more detail below.

As seen in Figure 8, the Kepler empirical shot noise floor in red is significantly lower than the TESS floor in green. The higher noise floor indicates that the magnitude of irreducible noise in the TESS data is significantly higher than that of the Kepler. This revelation was not surprising based on the architectural differences between the telescopes and was one of the main questions that this research wanted to address. The Kepler mission telescope has a focusing mirror area of 0.849 m^2 while the TESS mission telescope's total area is only 0.032 m^2 , over an order of magnitude smaller. The mirror's size directly affects the number of photons collected by the CCD sensor. The smaller mirrors in the TESS telescope lead to a lower and much less desirable signal-to-noise (SNR) ratio in the light curve data.

The second observation is that most observed TESS flicker is concentrated in a narrow band just above the shot noise floor, indicating that the shot noise overwhelms the granulation variation in the signal we are attempting to isolate. Based on flicker magnitude and ranges in published Kepler data, we expect that after subtracting the shot noise from the TESS data, the remaining residuals will display minimal variation due to granulation-based flicker. In fact, we surmise that much of the remaining residual contains additional shot noise not accounted for in our empirical shot noise removal methodology.

A key question this work is trying to answer is if it would still be possible to use TESS data for flicker-based gravity estimation. Based on the observations discussed above, it is highly likely that the SNR of the TESS data is too low for accurate modeling of surface gravity using the flicker method.

3. GRAVITY MODEL

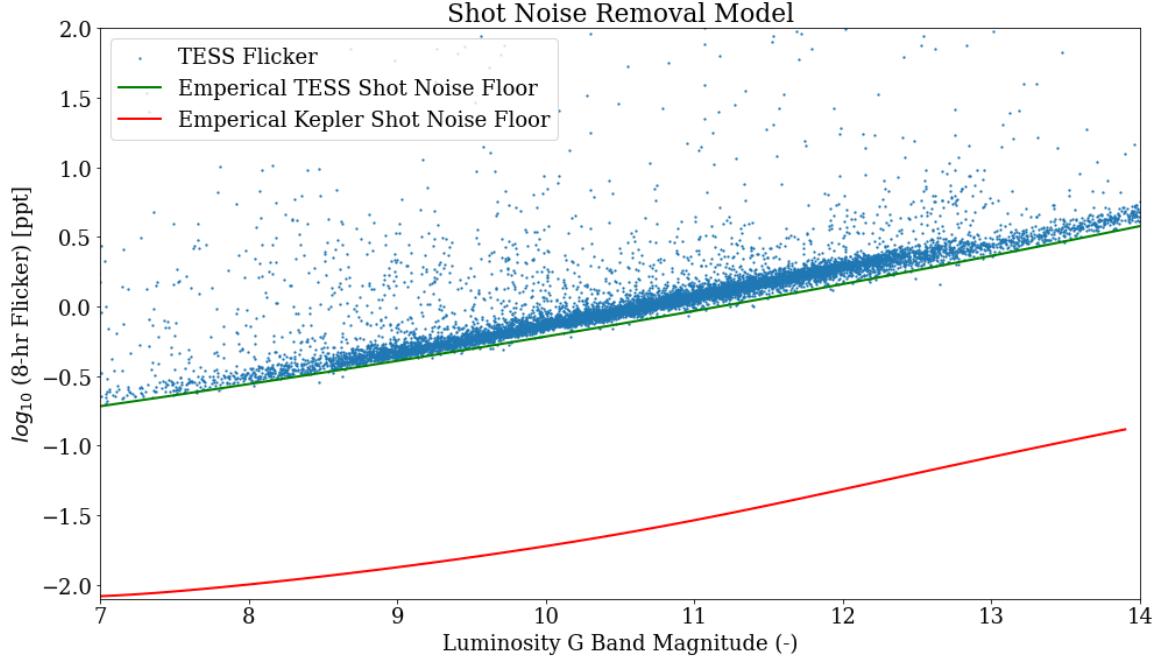


Figure 8. Shot noise floor calculation for TESS light curve data compared to equivalent curve calculated for Kepler data Bastien et al. (2016). TESS shot floor is considerably higher than the Kepler resulting in a lower signal-to-noise ratio due to differences in telescope architecture.

This section describes the process used to build models of surface gravity based on the data processing pipeline discussed in Section 2. The final shot noise corrected F_8 flicker from the data processing pipeline is used to create regression models based on gravity labels cross-matched to estimations from an alternative data source. We split the data into training, validation, and testing data sets and fit and tune linear regression, support vector regression, and LOWESS smoothing models. A final model is chosen based on performance on the validation data set and tested for a final estimate of accuracy on the test data set. All steps are described below in more detail.

3.1. Gravity Labels

In order to create regression models of gravity, we first need to obtain labels for the TESS stars of interest in this work. Gravity labels were created by cross-matching the stars in the TESS data set to surface gravity estimates in the recently available Gaia Release 3 database (Gaia & Bono 2016; Brown et al. 2021). These gravity estimations are based on stellar luminosity and radius models, which are independent of the flicker-based methodology discussed in this work and have numerous uncertainties associated with them. The astrophysical expectation is that since giant stars have lower surface gravity, they should show larger short-term variability in the form of the 8-hr flicker (Bastien et al. 2016). Note that for the remainder of this report, all analysis is conducted using \log_{10} gravity, as measured in cgs units, and is called gravity for brevity.

Cross-matching the TESS flicker data to the Gaia gravity labels results in a dataset of 9,736 stars which contain both flicker and gravity information. Additionally, we filter the stars to remove data that fall outside reasonable boundaries where the physics used to inform this model is expected to be valid (i.e., log gravity values between 0 and 5, corresponding to the expected range from “super giant”

down to “sub-dwarf” stars). Therefore we only model stars within the following ranges, resulting in a final count of 6,933 observations:

- Gravity (cgs): 0.0 -5.0
- Flicker (F_8): 0.0-5.0
- Temperature (K): 4500-6500

A plot of the cross-matched surface gravity as a function of F_8 flicker is shown in Figure 9. For this figure, we color the data as a function of G-band luminosity, or brightness, where lower values equate to brighter stars. The shape of the scatter plot is relatively flat and noisy compared to what we expect based on the physics and previously published Kepler data would suggest. We believe the discrepancies directly result from the low SNR ratio observed in the shot removal section. As previously suggested and now supported by the relationship plotted in Figure 9, it is likely that this data is not useful for accurately predicting surface gravity based on flicker. Despite this observation, we proceed with modeling efforts to develop the techniques, methods, and pipeline for future work.

We observe a cluster of outlier stars in the lower left-hand side of Figure 9 within the gravity and flicker range of 2-3 and 0.0-0.5. Of particular note, the stars in this cluster have low G-band Luminosity, which equates to brighter stars. Given the high shot noise floor determined in Section 2.8 and an understanding of the physics behind the plot, it is likely that the stars in the outlier cluster contain actual flicker-based information. The stars outside the cluster only encode information of shot noise and are not helpful for modeling surface gravity. Although outside this project’s scope, this analysis may be redone, focusing on only downloading and processing data for stars within the brightest G-band luminosity.

3.2. *models*

This section discusses the techniques employed to fit a model for estimating surface gravity. As discussed above, the usefulness of the current data is questionable due to the low signal-to-noise ratio and will likely not result in meaningful results. However, we implement the following steps to develop the modeling pipeline for future work.

The 6,933 observations are broken down into training, validation, and testing sets with an 80/20/10% respective split. The training data set is used to tune and select the best hyper-parameters for each model via 5-fold cross-validation, using mean squared error (MSE) as the model performance metric. The validation set is a holdout set used to evaluate the best tune of each model against each other. We use the test set for estimating the final selected model’s performance after it is refit to a combination of the training and validation data set.

3.2.1. *Polynomial Regression*

Polynomial regression is a type of regression modeling that uses a n^{th} degree polynomial to represent the relationship between the dependent and independent variable, where a 1^{st} degree polynomial is simple linear regression. A 3^{rd} order polynomial fit was used to model surface gravity in previously published work for the Kepler mission and is therefore included in this work [Bastien et al. \(2013\)](#). For this work, we consider polynomial orders between 1 and 3 and fit the models using the SciKit-Learn python software kit [Pedregosa et al. \(2011\)](#).

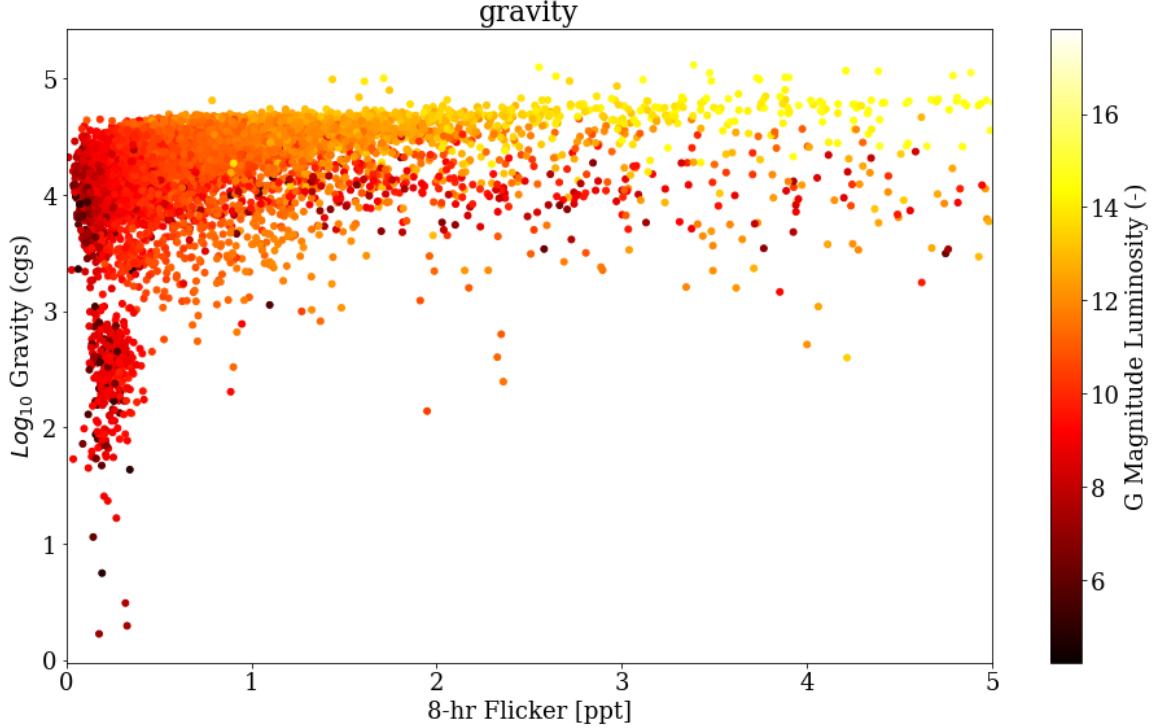


Figure 9. Surface gravity plotted as a function of F_8 flicker colored by G-band luminosity. The cluster of stars in the plot’s lower left-hand portion represents bright stars (low G-band Luminosity). Although these observations appear to be outliers, due to the low SNR of the TESS data, they are likely the only observations that contain actual flicker information as opposed to residual shot noise.

3.2.2. LOWESS

LOWESS smoothing combines weighted least squares regression and the k-nearest neighbor clustering algorithm. At every point in a data set, a low-order polynomial is fit with k nearest neighbor points, which are weighted based on their distance from the point of interest. The weights are assigned based on the tri-cube weight

$$w_i = \begin{cases} \left(1 - \left|\frac{x_i - x}{d_k}\right|^3\right)^3 & \text{if } |x_i - x| \leq d_k \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

for training data $i = 1, \dots, n$ and where $d_k =$ the k^{th} smallest distance from any x_i to x .

There are several advantages and disadvantages to LOWESS smoothing. One advantage is that LOWESS is a non-parametric model that does not require a global function specification of any form to provide a fit, as it does not fit a closed-form solution. A disadvantage is that it is computationally expensive and requires a lot of densely sampled data to provide a good fit. It is therefore not the best choice for high-dimensional data. The work presented in this report only uses a relatively small, 1-dimensional data set and is therefore a good candidate for LOWESS.

We consider two tuning parameters for the LOWESS model: the number of polynomial fits and the fraction of data. The number of fits specifies how many local polynomial fits are created, and the fraction of data determines how many data points around the anchor point the polynomial fits

include. An exhaustive grid search of the tuning parameters listed in Table 1 is used to determine the best hyperparameter settings for this data. The ‘MOE-Py’ library is used in this analysis for the LOWESS smoother (Bourn 2021).

Hyper Parameter	Values
Number of fits	1, 2, 3, 4, 5, 6, 7, 9
Fraction	0.2, 0.4, 0.6, 0.8, 0.9

Table 1. Tuning parameters for the LOWESS model.

3.2.3. Support Vector Regression

The SVM is a supervised learning method that works well with a small number of observations and is comparatively fast. The SVM works by maximizing the margin of hyperplanes used to separate linearly separable data. The linear SVM can be trivially extended to a non-linear case by use of kernels and the kernel trick (Hastie et al. 2001). Furthermore, the classic SVM, typically associated with classification problems, can be extended to support vector regression (SVR). The main idea of the SVR is similar to that of the SVM, where the goal is to minimize the error by determining the hyperplane that maximizes the margin given some tolerance for error.

We build the SVR models using the SciKit-Learn library in python (Pedregosa et al. 2011). Three kernels are considered in this work, each having its own specific set of tuneable hyper-parameters. An exhaustive grid search of the tuning parameters listed in Table 2 is conducted for each kernel.

kernel	C	Degree
linear	.01, .1, 1, 5	-
poly	.01, .1, 1, 5	2, 3
rbf	.01, .1, 1, 5	-

Table 2. Tuning parameters for the support vector regression (SVR) model

3.2.4. Results

All three models were tuned using 5-fold cross-validation of the training data set with MSE as the performance metric. The hyper-parameter combination that resulted in the lowest MSE is used to fit a model on the validation test set. The final selected hyper-parameters, the cross-validated MSE, and validation MSE are shown in Table 3. At least one of the final hyperparameters for all three models resides at our search grid’s limits. The selection of these bounding values indicates that the models prefer more flexibility given the function of these particular hyper-parameters. However, this is likely a result of excess shot noise in the data due to low SNR. Based on the governing physics and previously published work, we expect a less flexible model to perform better on non-noisy data than this analysis suggests and do not recommend a more expansive grid search at this time.

The 5-fold CV MSE indicates that the polynomial model performs the best, followed by the SVR and the LOWESS models. However, this is not a good indication of actual performance, and models should be selected using the validation test set. The tuned models are refit on the validation set, and

Model	Hyper-parameters	5-fold CV MSE	Validation MSE
Polynomial	Order = 3	12.93	14.02
LOWESS	frac = 0.4, Numb. Fits= 9	13.60	14.53
SVR	kernel = rbf, C = 5	13.55	14.65

Table 3. Final selected tuning parameters for all models.

the MSE is calculated. In this case, the polynomial performs the best, followed by the LOWESS and SVR models.

Figure 10 shows a plot of each model fit to the validation dataset. Qualitative inspection of the plot indicates that the polynomial and SVR models are likely overfit to sparse and noisy data resulting from poor data quality. The LOWESS model seems to be less flexible and more representative of what the expected fit would be. However, this observation does not indicate that the LOWESS model would be the best choice with a higher SNR dataset since this is probably a result of the range of hyper-parameters in the grid searches. It is again important to point out that this modeling exercise has been completed on data with excessively low SNR and is completed as an exercise to develop the modeling pipeline for future work and not to choose a definitive solution moving forward.

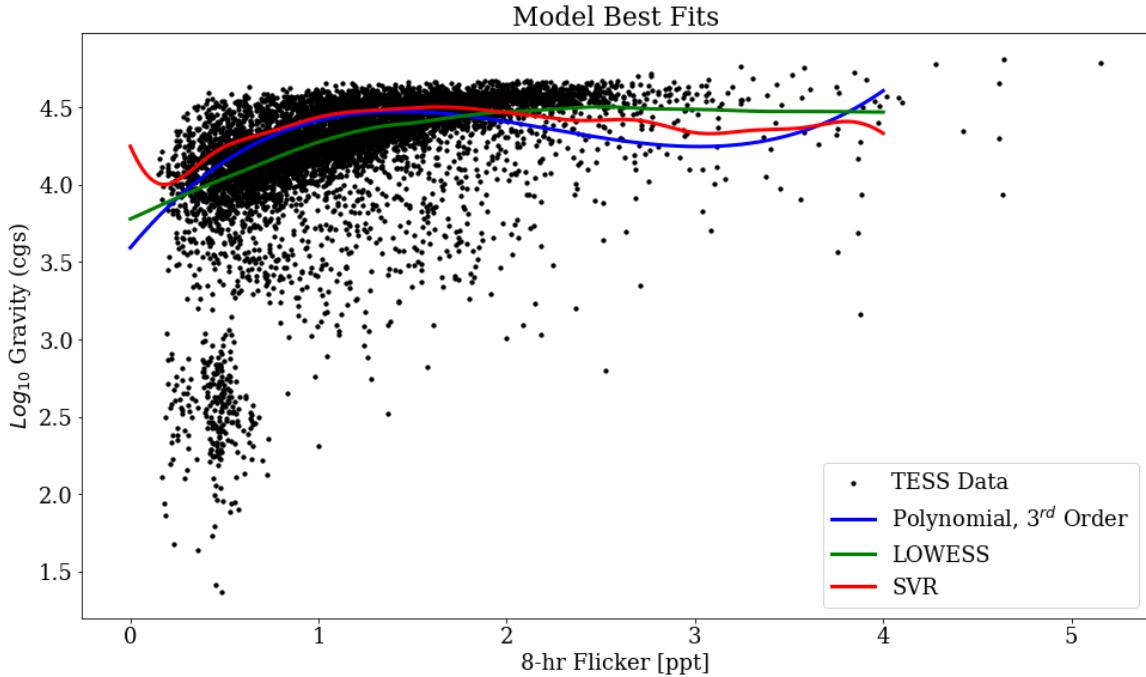


Figure 10. Model fit to the validation data using the best hyper-parameters as determined by 5-fold cross-validation on the training data set. The polynomial and SVR models display indications of overfitting to sparse, noisy data, resulting from the low SNR in the TESS data. The Lowess model displays a higher bias but is more likely to represent the expected natural behavior of good, high SNR data.

The final selected model for this particular data was the polynomial model. This selection is based on the validation MSE despite the obvious qualitative indications of overfitting, so we again note that this effort intends to develop the modeling pipeline for use on higher SNR data in future work. Finally, The 3rd order polynomial model was refit to the combined training and validation

datasets and used to predict the observations in the testing data set. The test MSE of the final model was 0.163. We observe that the MSE for the training, validation, and testing increases with each evaluation respectively. This behavior is expected and gives some confidence in the accuracy expectation for the final deployed model.

4. CONCLUSIONS

In this report, we discuss the development of a data processing and modeling tool that uses observed flicker in light curve data from the NASA TESS mission to estimate stellar surface gravity. Successful development of this methodology would allow astronomers and astrophysicists to quickly and accurately estimate stellar surface gravity using readily available data for many stars where previous gravity estimation methodologies lack accuracy, are expensive to perform, or rely on specific data that is not available. Although the flicker-based methodology has been successfully demonstrated on Kepler mission data, this is the first known application to TESS data, and it is not clear if its application is possible due to architectural differences between the Kepler and TESS telescopes.

The data processing pipeline developed in this work included numerous steps that needed optimization to run effectively and efficiently. These steps include acquiring the required data from various publicly available databases; preprocessing the data to remove unwanted trends, outliers, and noise; applying smoothing techniques to isolate and extract variance on time scales of interest; calculating the appropriate flicker; and training and evaluating various models for estimating surface gravity as a function of flicker. Each step in the data processing pipeline was tailored for the TESS data and optimized to run as fast as possible without decreasing quality. We investigate three machine learning models for the modeling portion of the pipeline: polynomial regression, support vector regression, and LOWESS models.

After completing the data processing portion of this project, it became apparent that the shot noise inherent to the TESS mission data is stronger than the flicker signal used for estimating surface gravity. Therefore flicker can not be used to estimate surface gravity for TESS data using the current methodology. This result is not wholly unexpected since the TESS telescope has a smaller mirror surface area than the Kepler telescope, which is known to impact the signal-to-noise ratio negatively. This result answers one of the critical questions this work was seeking to answer: if the methodologies used to model stellar gravity with Kepler data are directly applicable to the TESS data, given the differences in telescopes. We develop the modeling portion of this work using the low SNR data, acknowledging that the final models may not be useful in their current form, but the process and pipeline can easily be re-run with better data in the future.

Despite the initial conclusion that the TESS data is not suitable for estimating surface gravity using the current methodology, we have recognized numerous opportunities for future work. We have identified a cluster of outlying points in Figure 9 representing bright stars. These stars may have a high enough SNR to encode flicker information. This analysis may be re-done, focusing on a more extensive set of stars from outside the continuous viewing zone with similar G-magnitude values as those in the cluster of higher SNR TESS observations. An additional area of study for future work could be investigating if aggregation of the 2-minute observations into longer window fluxes can effectively increase the SNR, making the TESS data more useful. Finally, current published work and the work discussed in this report assumes homoscedastic behavior of the residual flicker. There is no current justification to assert that this assumption is valid, what effects heteroscedastic behavior

would have on the accuracy of gravity estimations if present, and if we can ascertain any additional information on the nature of the stars from studying the variation of standard deviation over time.

I would like to thank Dr. James Davenport from the Institute for Data Intensive Research in Astrophysics and Cosmology for his expertise, guidance, and encouragement throughout this project.

REFERENCES

- Aerts, C., Christensen-Dalsgaard, J., & Kurtz, D. W. 2010, *Asteroseismology* (Springer Science & Business Media)
- Bastien, F. A., Stassun, K. G., Basri, G., & Pepper, J. 2013, *Nature*, 500, 427
- . 2016, *The Astrophysical Journal*, 818, 43
- Bourn, A. 2021, *moeipy*,
doi: [10.5281/zenodo.4642896](https://doi.org/10.5281/zenodo.4642896)
- Brown, A. G., Vallenari, A., Prusti, T., et al. 2021, *Astronomy & Astrophysics*, 649, A1
- Brown, T. M., Gilliland, R. L., Noyes, R. W., & Ramsey, L. W. 1991, *The Astrophysical Journal*, 368, 599
- Caldwell, D. A., Tenenbaum, P., Twicken, J. D., et al. 2020, *Research Notes of the AAS*, 4, 201
- Chaplin, W. J., Kjeldsen, H., Christensen-Dalsgaard, J., et al. 2011, *Science*, 332, 213
- Gaia, C., & Bono, G. 2016
- Ghezzi, L., Cunha, K., Smith, V., et al. 2010, *The Astrophysical Journal*, 720, 1290
- Hastie, T., Tibshirani, R., & Friedman, J. 2001, New York, NY, USA
- Huber, D., Bedding, T., Stello, D., et al. 2011, *The Astrophysical Journal*, 743, 143
- Isaac. 2022, Tess Observations.
<https://tess.mit.edu/observations/>
- Kjeldsen, H., & Bedding, T. R. 2011, *Astronomy & Astrophysics*, 529, L8
- Marston, A., Hargis, J., Levay, K., et al. 2018, in *Observatory Operations: Strategies, Processes, and Systems VII*, Vol. 10704, SPIE, 416–428
- Mathur, S., Hekker, S., Trampedach, R., et al. 2011, *The Astrophysical Journal*, 741, 119
- Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, *Journal of Machine Learning Research*, 12, 2825
- Ricker, G. R., Winn, J. N., Vanderspek, R., et al. 2014, *Journal of Astronomical Telescopes, Instruments, and Systems*, 1, 014003, doi: [10.1117/1.jatis.1.1.014003](https://doi.org/10.1117/1.jatis.1.1.014003)
- Stello, D., Huber, D., Bedding, T. R., et al. 2013, *The Astrophysical Journal Letters*, 765, L41
- STSI. 2022, Tess Coverage. <https://archive.stsci.edu/missions-and-data/tess>
- Twicken, J. D., Caldwell, D. A., Jenkins, J. M., et al. 2020
- Valenti, J. A., & Fischer, D. A. 2005, *The Astrophysical Journal Supplement Series*, 159, 141
- Van Rossum, G., & Drake, F. L. 2009, *Python 3 Reference Manual* (Scotts Valley, CA: CreateSpace)