Progress Report*

Yufei Liu

2025-03-03

1 Introduction

Stellar flares are sudden, intense increases in the brightness of stars, caused by magnetic reconnection events in their atmospheres. These energetic outbursts can provide valuable insights into stellar activity and characteristics of the stars. The Transiting Exoplanet Survey Satellite (TESS) from NASA provides high-precision light curve data that can be used to detect and analyze these flares.

In this study, we explore flare detection methods using TESS light curve data. Our approach focuses on anomaly detection methods, given the unique characteristics of flares as deviations from the typical brightness variations of a star. Specifically, we investigate time series models, DBSCAN, Gaussian Mixture Models (GMM), and Isolation Forest to detect flares based on residual and outlier identification, and density-based clustering. This study aims to assess the feasibility of unsupervised learning for stellar flare detection and evaluate the performance of different methods in identifying transient stellar events.

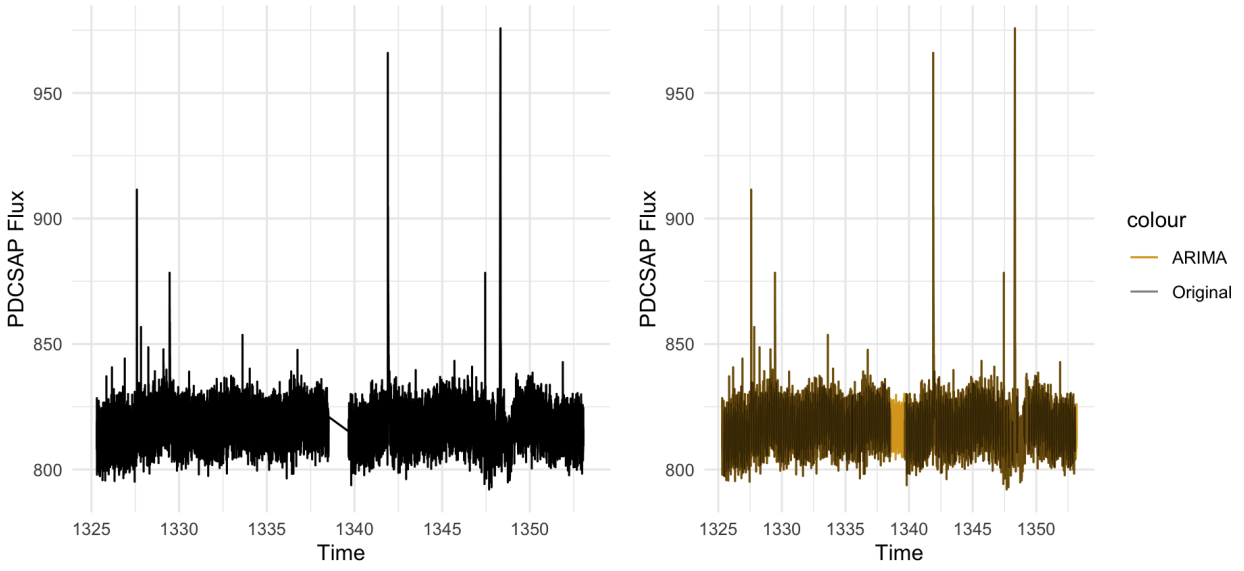
2 Data

To explore different anomaly detection approaches, we use light curve data for star TIC 129646813 from TESS mission, which is available on the MAST website. The dataset consists of time-series observations of stellar flux, where each observation captures the brightness of a star over time. These observations are taken at regular intervals, typically every 2 minutes (short-cadence mode). We use the Pre-search Data Conditioned Simple Aperture Photometry (PDCSAP) flux to conduct our analysis since it is clearer and contains less noise due to detrending manipulations. Flares appear as transient increases in flux, often characterized by a rapid rise followed by a gradual decay. In this study, we preprocess the data to handle missing values and standardize the flux measurements before applying unsupervised learning methods to identify potential flare events as anomalous deviations from the expected stellar variability.

As shown in Figure 1a, there is a gap in time between 1338 and 1340. Since the data was collected over a regular time interval, such gap may due to instrumental errors. Also, there exists a few missing values in flux at the end of the time series. Since for some methods we are interested in, such as ARIMA and DBSCAN, the algorithm could not handle missing value naturally, thus we need to either remove or impute the missing values. As shown in Figure 1b, the yellow line represents the imputation result using ARIMA interpolation in `imputeTS` package in R. It fills both the gap in time and missing values in flux.

When imputing the missing values, we assume that the behaviour of the star's flux has the same pattern as it was before the gap and after the gap However, to validate the necessity of imputation, the following analysis will compare the model results for both imputed data and original data.

*Code and data are available at: https://github.com/Florence-Liu/Stellar-Flare



(a) original

(b) with imputation

Figure 1: Light curve time series for TIC 129646813

3 Methods

3.1 Time Series Analysis

To analyze the variability in the light curve data and identify potential flare events, we first model the time series using an autoregressive integrated moving average (ARIMA) approach, which could capture the temporal dependencies in the data.

An ARIMA(p, d, q) model consists of three components:

- **Autoregressive (AR) term (p)**: The current value of the time series is expressed as a linear function of its previous values, capturing persistent trends.
- **Integrated (I) term (d)**: Differencing is applied to the time series d times to achieve stationarity, meaning that statistical properties (such as mean and variance) remain constant over time.
- **Moving Average (MA) term (q)**: The model includes a linear combination of past forecast errors to account for short-term fluctuations.

Mathematically, an ARIMA(p, d, q) model is defined as:

$$\Phi_p(B)(1 - B)^d Y_t = \Theta_q(B)\epsilon_t, \tag{1}$$

where:

- B is the backshift operator, such that $BY_t = Y_{t-1}$,
- $\Phi_p(B)$ is the autoregressive polynomial of order p,
- $\Theta_q(B)$ is the moving average polynomial of order q,
- ϵ_t is a white noise error term with mean zero and constant variance.

We apply the `auto.arima` function to find the best fit model based on Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC). After fitting the model, we examine the residuals and identify outliers in the residuals using a 3-sigma criterion, where points deviating more than three standard deviations from the mean residual are flagged as anomalies.

3.2 Machine Learning Methods

3.2.1 Isolation Forest

To further identify potential stellar flares in the TESS light curve data, we apply an unsupervised anomaly detection method known as the Isolation Forest (IF). The Isolation Forest method is based on recursive partitioning of the data, where an ensemble of randomly constructed binary trees is used to isolate data points. It is based on the assumption that because anomalous points, which are rare and significantly different from the majority, they can be isolated using few partitions.

Given a dataset $X = \{x_1, x_2, \ldots, x_n\}$ with d features, the Isolation Forest algorithm operates as follows:

1. Select a subset of the data and repeat randomly splitting the data on randomly chosen features until each instance is isolated.
2. A collection of isolation trees (iTrees) is built, where each tree represents a different partitioning of the data.
3. The anomaly score of a data point x is computed based on the average path length required to isolate x across all trees. The anomaly score is defined as:

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}} \tag{2}$$

   where:

   - $E(h(x))$ is the expected path length of x,
   - $c(n)$ is the average path length of an unsuccessful search in a binary tree of size n.

   Data points with high anomaly scores are flagged as potential outliers.

We use the `IsolationForest` function fro `sklearn` library in Python to fit the time series data and visualize the anomaly detection results. For IF, the algorithm could handle the missing values and irregular time space, so we don't need furthur manipulation on data.

3.2.2 Gaussian Mixture Model

3.3 Anomaly Detection with Gaussian Mixture Model

To identify stellar flares in the TESS light curve data, we apply the *Gaussian Mixture Model* (GMM), a probabilistic model that represents the distribution of the data as a mixture of multiple Gaussian components. GMM is widely used for clustering and anomaly detection, as it provides a flexible way to model complex distributions.

3.3.1 Gaussian Mixture Model Formulation

The GMM assumes that the observed data $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ follows a mixture of K Gaussian distributions, each defined by its mean and covariance. The probability density function (PDF) for a given data point x is given by:

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k), \tag{3}$$

where:

- K is the number of Gaussian components in the mixture,
- π_k represents the mixing weight of the k-th Gaussian, subject to $\sum_{k=1}^{K} \pi_k = 1$,

- $\mathcal{N}(x \mid \mu_k, \Sigma_k)$ is the Gaussian distribution with mean μ_k and covariance matrix Σ_k:

$$\mathcal{N}(x \mid \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)\right). \tag{4}$$

The parameters of the GMM (π_k, μ_k, and Σ_k) are estimated using the *Expectation-Maximization* (EM) algorithm, which iteratively refines the parameters by maximizing the likelihood of the observed data.

3.3.2 Implementation and Outlier Detection

For this study, we apply the GMM to the TESS light curve data, treating flux values as observations to be modeled as a mixture of normal distributions. The steps involved are:

- **Preprocessing:** The flux values are normalized to ensure consistency in clustering.
- **Model Fitting:** A GMM is trained with varying numbers of components K, selecting the optimal number using the Bayesian Information Criterion (BIC) or Akaike Information Criterion (AIC).
- **Anomaly Scoring:** The likelihood of each data point is computed under the fitted GMM, where lower likelihood values indicate deviations from the learned distribution.
- **Thresholding:** A threshold is determined empirically, and points with the lowest likelihoods are classified as anomalies.

By visualizing the probability density function and comparing flagged anomalies with those identified using ARIMA and Isolation Forest, we evaluate the effectiveness of GMM in detecting stellar flares. Since flares are transient increases in brightness, they are expected to belong to a low-probability region of the modeled flux distribution, making GMM a suitable approach for anomaly detection in this context.

3.3.3 DBSCAN

3.4 Anomaly Detection with DBSCAN

To identify stellar flares in the TESS light curve data, we apply the *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN) algorithm. DBSCAN is a clustering method that groups data points based on their density and can effectively detect anomalies as points that do not belong to any dense cluster. This makes it well-suited for detecting flares, which are expected to be rare and significantly different from normal stellar variability.

3.4.1 DBSCAN Algorithm

DBSCAN classifies data points into three categories:

- **Core points**: Points that have at least a minimum number of neighbors within a specified distance.
- **Border points**: Points that are within the neighborhood of a core point but do not meet the density criteria themselves.
- **Noise (Outliers)**: Points that do not belong to any cluster, often representing anomalies.

The algorithm operates as follows:

1. **Neighborhood Identification**: For each point x_i, count the number of points within a predefined distance ϵ.
2. **Cluster Formation**: If x_i has at least $MinPts$ neighbors, it is a *core point*, and a new cluster is formed.
3. **Density Expansion**: The cluster is expanded by recursively adding reachable core and border points.
4. **Outlier Detection**: Points that are not assigned to any cluster are classified as anomalies.

3.4.2 Implementation and Outlier Detection

For this study, we apply DBSCAN to the TESS light curve data, treating flux values as input for clustering and anomaly detection. The implementation consists of the following steps:

- **Preprocessing:** The flux values are standardized to ensure consistent distance measurements.
- **Hyperparameter Selection:** The parameters ϵ (neighborhood radius) and $MinPts$ (minimum points required for a core point) are chosen empirically or using techniques such as the k-distance plot.
- **Clustering:** DBSCAN is applied to group similar flux values and identify low-density regions as outliers.
- **Anomaly Classification:** Points labeled as noise (outliers) by DBSCAN are flagged as potential stellar flares.

By comparing DBSCAN-identified outliers with anomalies detected using ARIMA, Isolation Forest, and Gaussian Mixture Models, we evaluate the algorithm's effectiveness in identifying transient flux variations. Since flares appear as sudden brightness spikes, they are expected to reside in low-density regions, making DBSCAN a useful method for detecting them.

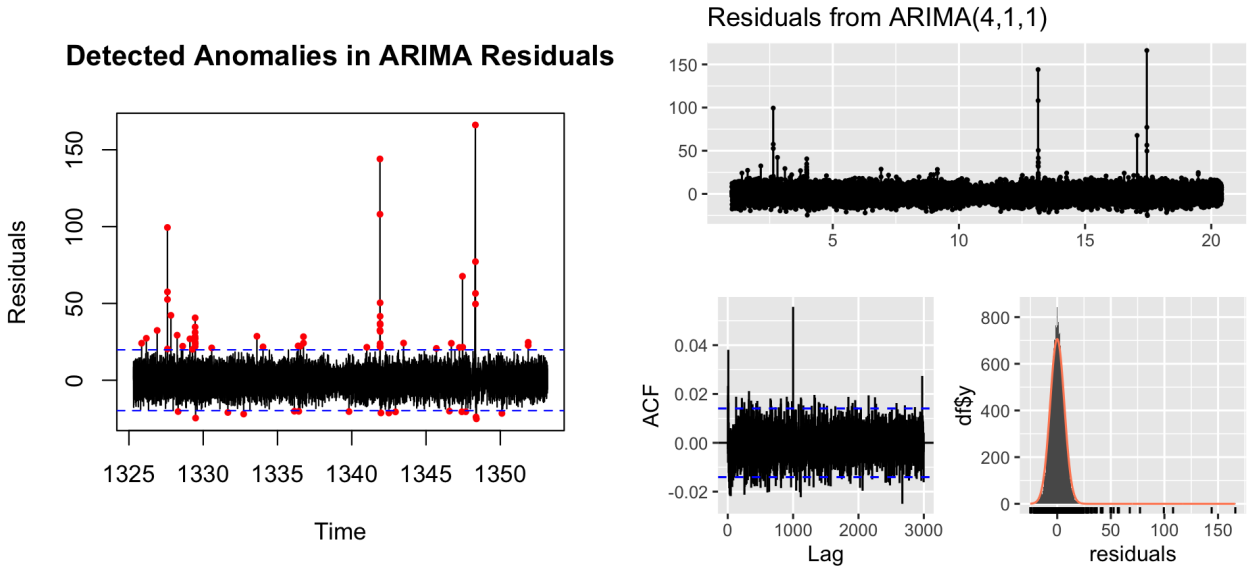
4 Results

4.1 Time Series Model

For this study, we apply the `auto.arima` function to select an optimal ARIMA model based on the Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC). The selected model for the light curve data is ARIMA$(4, 1, 1)$, which indicates:

- $p = 4$: Four lagged terms are included to account for autocorrelation.
- $d = 1$: Differencing is applied once to remove trends and ensure stationarity.
- $q = 1$: One lagged error term is included to model short-term fluctuations.

4.1.1 With Missing Value Imputation

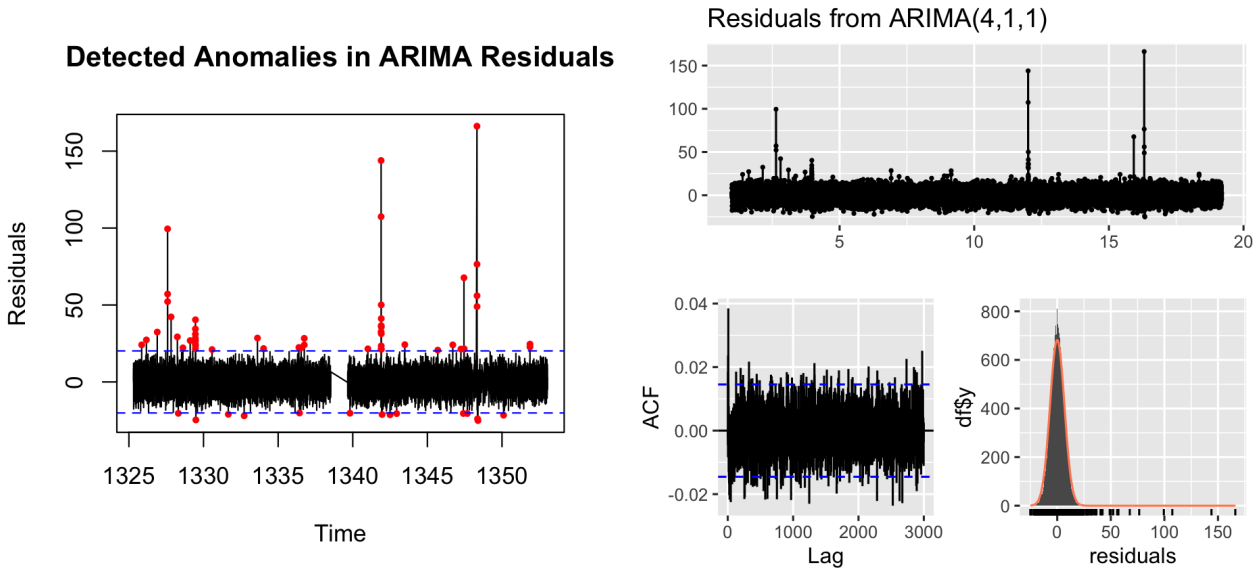


(a) result

(b) residual

Figure 2: model results

4.1.2 Without Missing Value Imputation



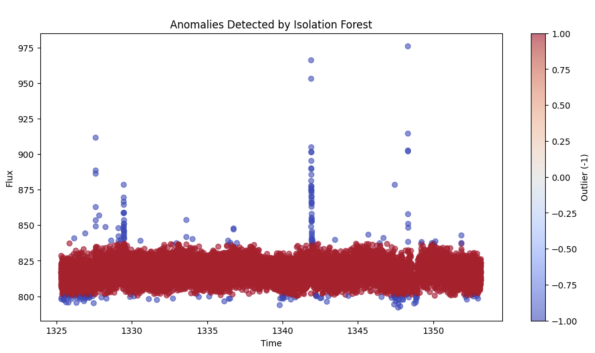
(a) result

(b) residual

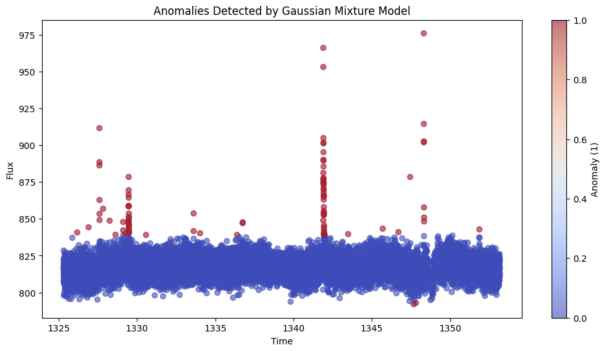
Figure 3: model results

4.2 Machine Learning Model

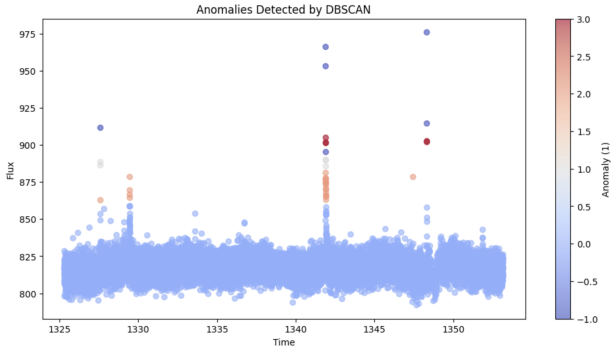
4.2.1 With Missing Value Imputation



(a) if



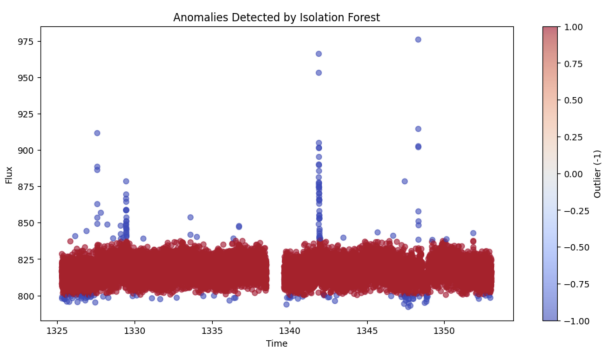
(b) gmm



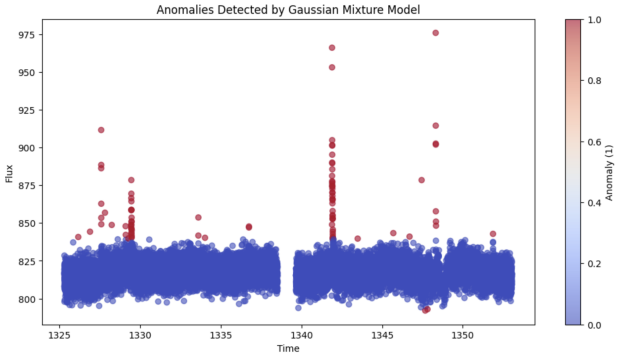
(c) dbscan

Figure 4: model results

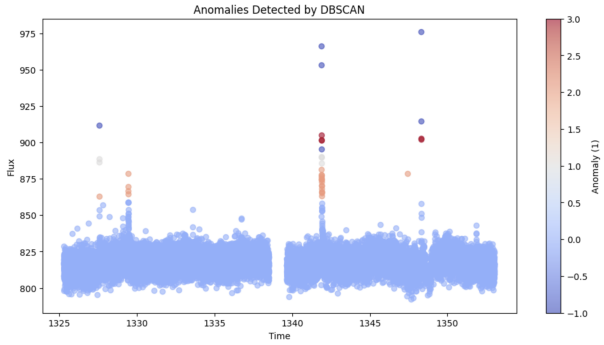
4.2.2 Without Missing Value Imputation



(a) if



(b) gmm



(c) dbscan

Figure 5: model results

4.3 Comparison

5 Discussion

5.1 Current Work

decide to use no imputation data, why sensitivity analysis

5.2 Next Steps

Tune parameter

decide validation and metric to compare model performance