



KRITTIKA SUMMER PROJECTS 2022

Stellar Flares

Abhinna Sundar, Ashwin Goyal, Chirag Verma,
Gourav Kumawat, Sandeepan Naskar, Sarthak
Talukdar, Devansh Jain, and Aravind Bharathi

KRITTIKA SUMMER PROJECTS 2022

Stellar Flares

Abhinna Sundar¹, Ashwin Goyal², Chirag Verma³, Gourav Kumawat⁴,
Sandeepan Naskar², Sarthak Talukdar⁵, Devansh Jain², and Aravind
Bharathi²

¹Indian Institute of Science Education and Research (IISER), Mohali - 140306, India

²Indian Institute of Technology (IIT) Bombay, Mumbai - 400076, India

³Indian Institute of Technology (IIT) Roorkee, Uttarakhand, 247677, India

⁴Indian Institute of Science Education and Research (IISER), Bhopal 462066, India.

⁵Indian Institute of Science, Bangalore 560012, India

Copyright © 2022 Krittika IITB

PUBLISHED BY KRITTIKA: THE ASTRONOMY CLUB OF IIT BOMBAY

[GITHUB.COM/KRITTIKAIITB](https://github.com/KrittikaIITB)

First Release, September 2022

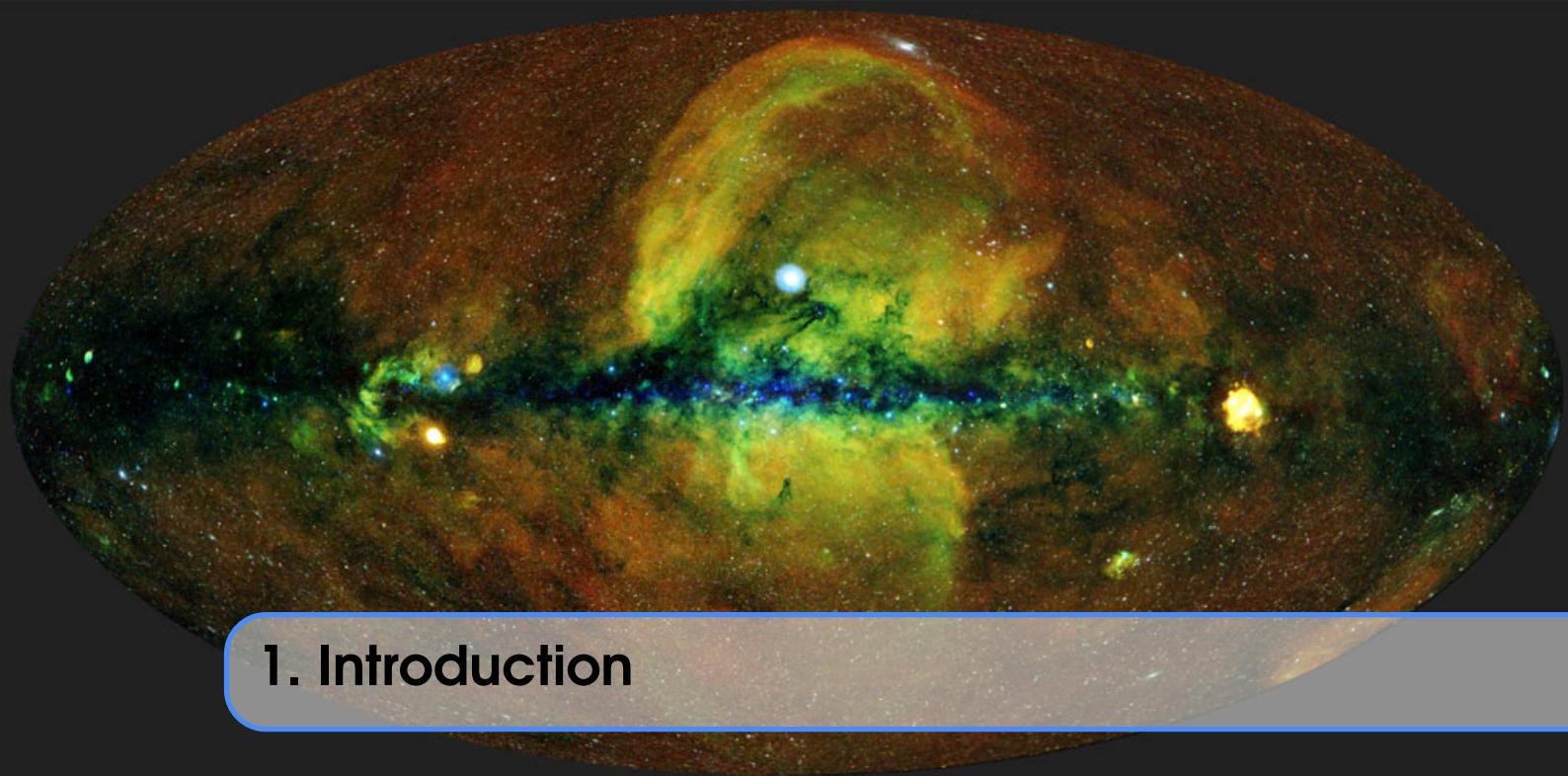
Abstract

We replicated the earlier models for the purpose of explaining a variety of stellar flares. We also provided our own amendments and potential issues with the existing models. We assumed that a flare consists of two components, the first of which rises in brightness quickly and relatively strongly, the second of which is slower and fainter. The latter is the consequence of the star's photosphere reflecting some of the energy from the first flare. The flare's location in relation to the stellar visible disc determines the flare's morphology, and the model is mostly based on geometrical considerations.



Contents

1	Introduction	4
1.1	Background	4
1.2	Observations and Datasets	5
1.3	Aims and Objectives	5
2	Data Cleaning	6
3	Trigger Algorithm	7
3.0.1	Constant Background fits	9
3.0.2	Linear and Polynomial Background fits	12
4	NOAA Algorithm	15
5	Local Extrema Algorithm	17
6	Results and Discussions	22
7	Conclusions and Future Works	23
8	Acknowledgements	24
	Bibliography	25



1. Introduction

1.1 Background

Flare activity often occurs or is constantly present in the early stages of stellar evolution. However, star formation theories essentially ignore this fact. The flare event could be crucial for comprehending star creation and early development. [Mirzoyan \(1981\)](#); [Parsamian and Chavira \(1969\)](#) Understanding the mechanism(s) of flare is complicated by the broad range of light curve morphologies. Specifying and predicting these light curves can help us learn important details about the solar processes that influence planetary ionospheres and thermospheres.

According to an examination of the light curves of flares recorded with a high time resolution, the vast array of distinct manifestations of these profiles may be divided into the following types. First type of profile includes a single flare where the brightness increases quickly for a brief period of time before declining slowly. Other type of profile consists of two or more distinct brightness peaks and are called the composite flares or double/higher order flares. Very often the decline of the first flare is followed by a second, fainter and much slower component, but in some cases only a slow rise and slow decline of brightness is observed. [Tovmassian et al. \(2003\)](#)

Elementary Flare Profile

We employed a straightforward mathematical formula (Equation eq1) that describes a so-called elementary soft X-ray flare time profile in order to understand specific flare brightenings seen on the light curve (EFP) [Gryciuk et al. \(2017\)](#). We parametrized simple, single-peaked events and break down complicated, multi-peaked flares into a sequence of basic events by fitting the profile to the data. The observed light curve's form was connected to the temporal profile of energy release. It is expected that the energy emitted at a particular moment would diminish quickly and monotonically according to a Gaussian distribution $g(x)$. The

dissipation of energy released is assumed to be a monotonically decreasing function $h(x)$. The energy dissipates (via radiation and conduction processes), and at time t (following x), the energy that is now accessible is $g(x)h(t-x)dx$. The integral of df from 0 to t represents the total energy $f(t)$ accessible at time t .

$$f(t) = \int_0^\infty g(x)h(t-x)dx \quad (1.1)$$

where $g(x) = Ae^{-(x-B)^2/C^2}$, and $h(x) = e^{-Dx}$.

We would be using this convolution function for modelling our flares as described as Method 1 in the subsection 3.

1.2 Observations and Datasets

The *Solar X-ray Monitor* (XSM), onboard *Chandrayaan-2*, is an instrument of the India's lunar mission launched on July 22, 2019. XSM continuously monitors the Sun and measures the Solar spectrum in the energy ranges of 1 – 15 keV with an energy resolution of ~ 175 eV at 5.9 keV. The spectral performance of the in-flight observations were found to be identical to that of the ground based observations. [Mithun et al. \(2020\)](#) Though the primary objective of the XSM was to provide the solar spectrum for the X-ray fluorescence spectroscopy experiment on the Chandrayaan-2 orbiter, which maps the elemental abundance of the lunar surface, but the XSM data can be used to independently study the Sun. These can be used to study Solar activity. It is designed in such a way that it carries out the flux measurements with time cadence of 1 sec over a wide range of solar activity, from below A-class to upto X-class X ray emission.

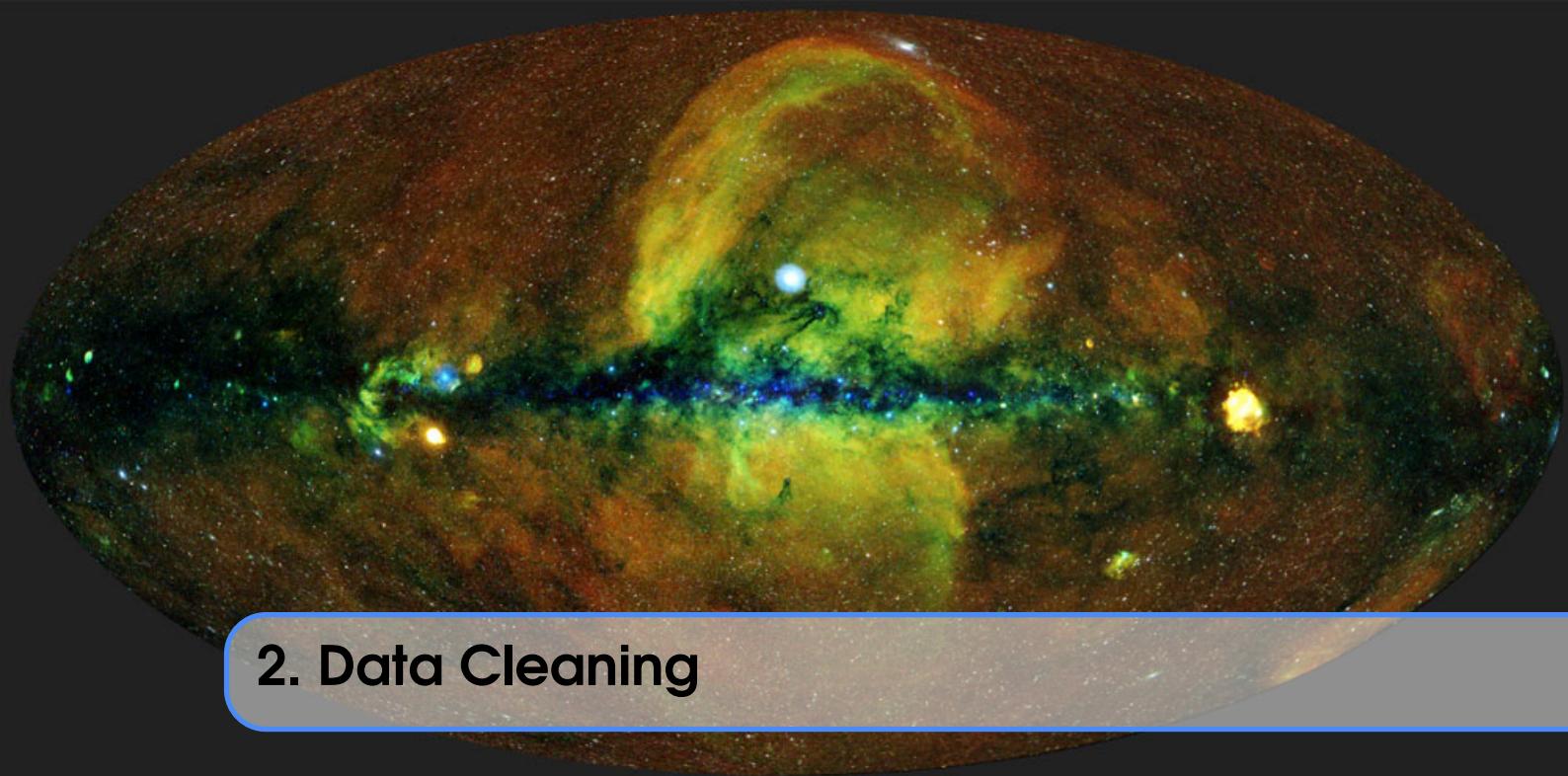
The raw level 1 data and the calibrated level 2 XSM data is organized day wise and can be downloaded from PRADAN portal ¹ of ISRO Science Data Archive (ISDA) at the Indian Space Science Data Center (ISSDC), Bangalore. We made use of the raw level 1 data, i.e. the .lc files for performing temporal analysis.

1.3 Aims and Objectives

The goal primarily comprises of detection, identification, and analysis of Solar Flares in X-Ray Light curves observed by XSM. This will help us in developing a robust and reliable pipeline for XSM temporal data. The XSM has more sensitivity than the GOES, and its detections are publicly accessible. By developing a code which is able to perform the aforementioned tasks, it was required to follow up those by implementing it to the GOES and the XSM data. It is also required to test whether the NOAA detection algorithm which is the current flare detection algorithm, used by NASA, and other algorithms best or not.

Apart from all these previously known detection and identification methods and algorithms of stellar flares, our task also consists of developing and implementing our own algorithm without diverting from the main goal, i.e. making a robust and efficient code. The analysis consists of fitting a good model to the detected flares. Additionally, those few flares which are not identifiable by the code, but could be identified visually, needs to be fitted using suitable models.

¹<https://pradan.issdc.gov.in/pradan/>

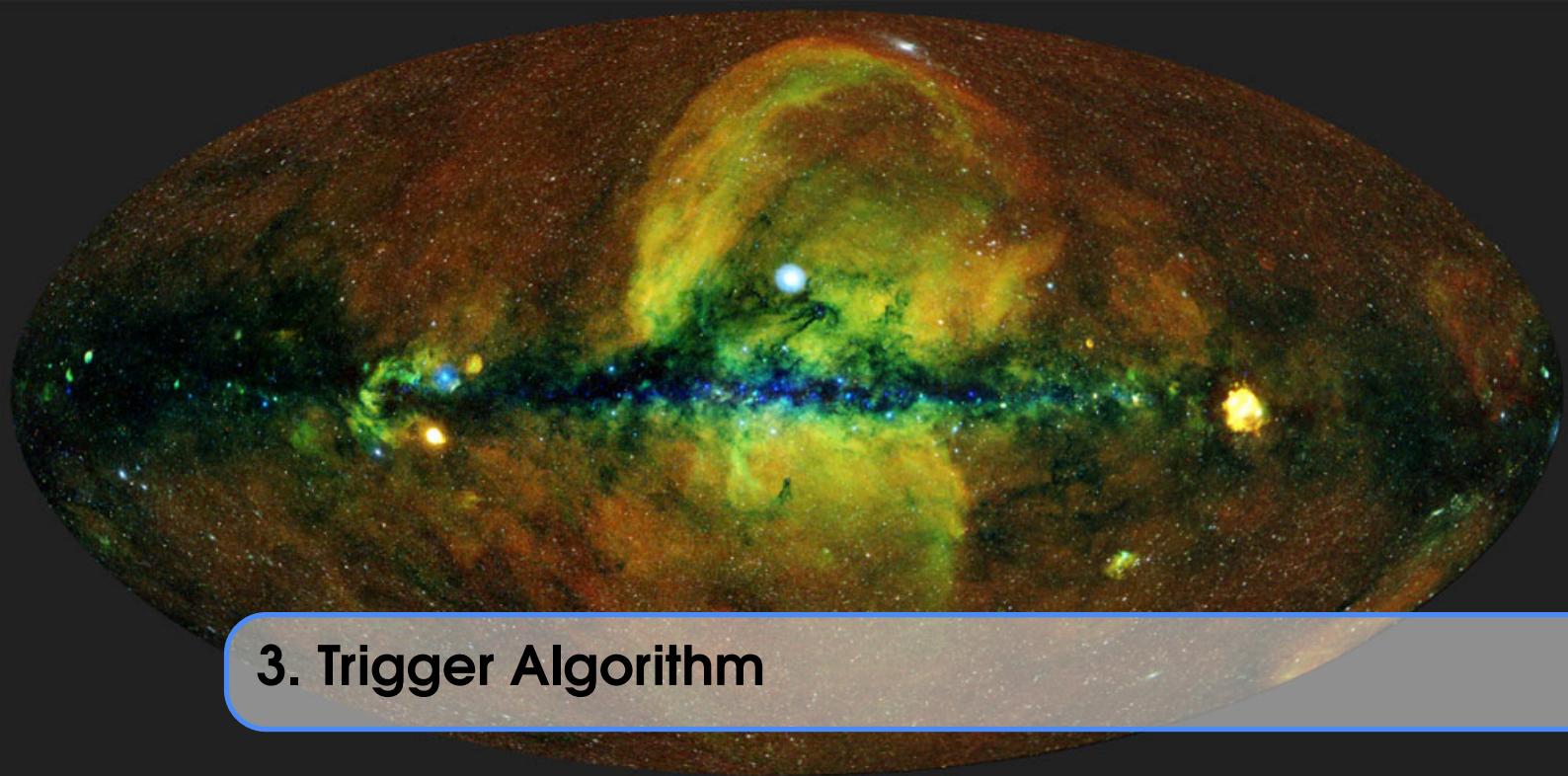


2. Data Cleaning

We tried implementing various methods as described in [Gryciuk et al. \(2017\)](#) and [Aschwanden and Freeland \(2012\)](#). Picking a few good plots with sufficient data points based on the noise in the data and manually identifying the peaks and deciding if they are flares in order to get familiar with the identification and detection process before automating the same. The following subsections include the different methodologies (modified version of the articles mentioned above) used by us to detect, identify and analyze the flares:

- The initial step involved the pre-processing of the raw data so as to convert the continuous count rate values to categories using the method of binning. Given our data was of one day in length i.e. 86400 s or 24 h. The binning was kept around 50 – 200 s which produced decent results.
- To clear out the roughness, reduce the noise, and to help us see better trends and patterns in the data, smoothing was done. This also helped in eliminating the outliers in the data. We used Boxcar averaging for smoothing our data, using Box1DKernel¹ available as Astropy python package [Robitaille et al. \(2013\)](#). Different days give different results based on the level of solar activity on that day.

¹<https://docs.astropy.org/en/stable/api/astropy.convolution.Box1DKernel.html>



3. Trigger Algorithm

After binning and smoothing, we tried detection, identification and analysis of the lightcurves using various methods. Here in this report, we will show our implemented method using 3 event files of XSM data i.e. data of 01 Oct 2019, 06 Apr 2020, and 07 Apr 2020.

- The algorithm for flare detection has been implemented almost directly from [Gryciuk et al. \(2017\)](#). First, we start by finding 4 consecutive increasing points such that the slope joining first and last point is more than a certain threshold. This threshold lies between 1.03 and 1.08. Then, we kept moving ahead and find 3 consecutive decreasing points. This means that the local peak must lie in between these points. We do this by finding the maxima between the initial and final set of points to determine the peak. It is also advisable to do a reverse pass after this. i.e to run the same algorithm in reverse and find other peaks. This is shown in the figures [3.1](#), [3.2](#), and [3.3](#) below.

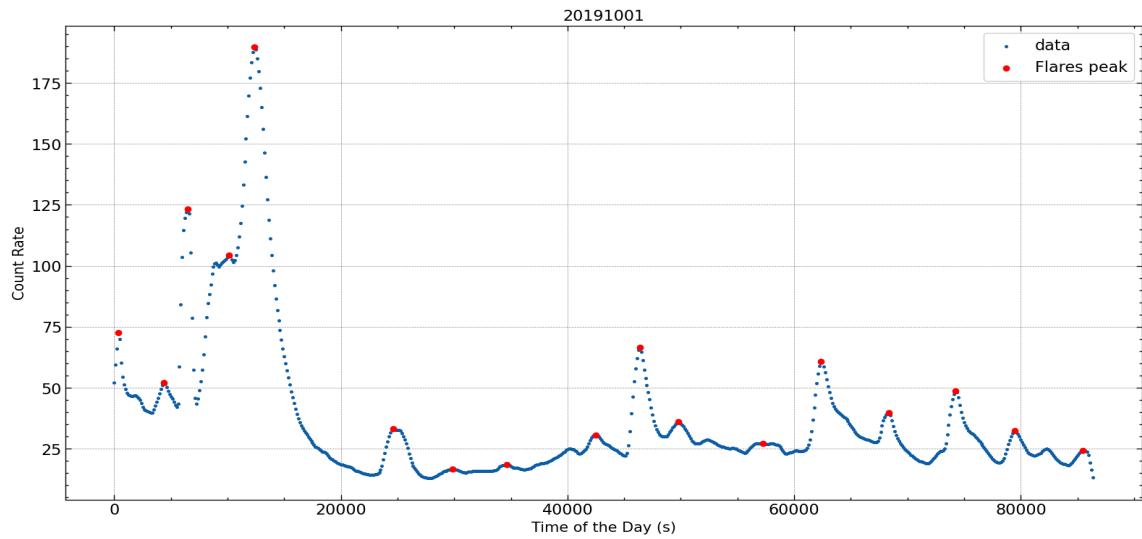


Figure 3.1: Light Curve with raw detected flare peaks in red (01 Oct, 2019)

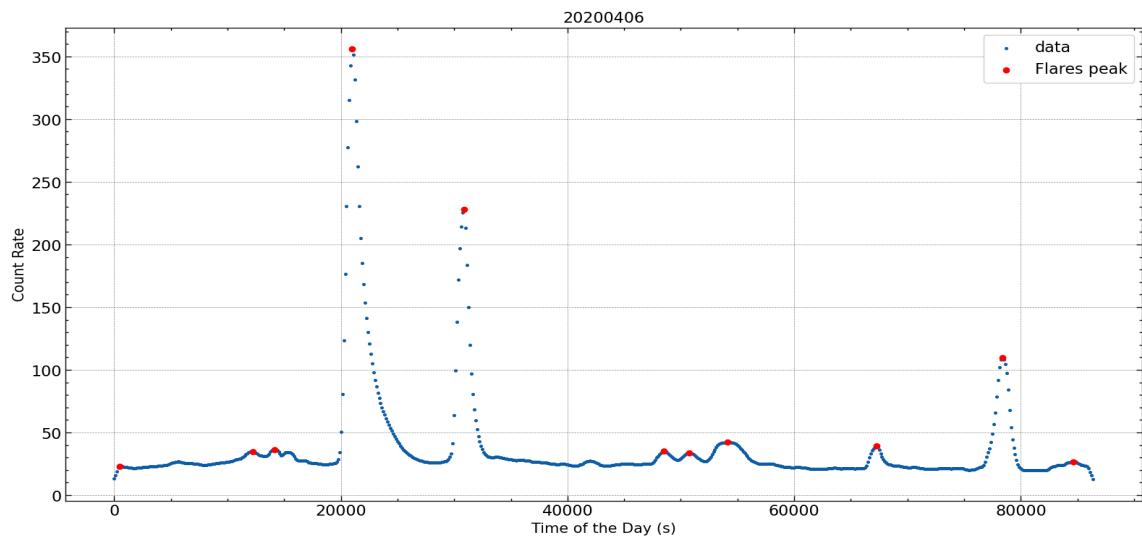


Figure 3.2: Light Curve with raw detected flare peaks in red (06 Apr, 2020)

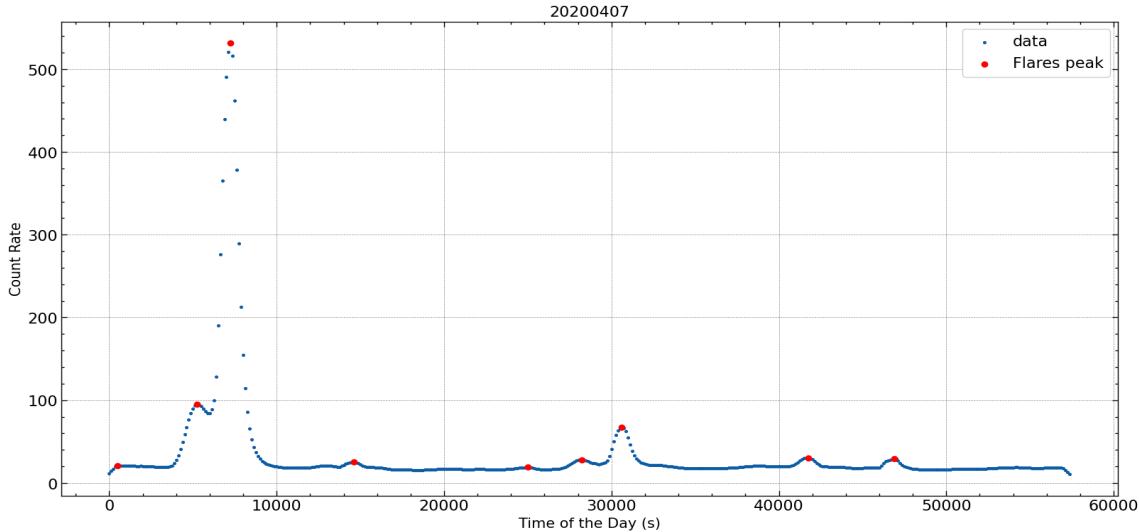


Figure 3.3: Light Curve with raw detected flare peaks in red (07 Apr, 2020)

- Once these raw peaks were detected, it was needed to be checked which peaks are real and which ones false detections. This required the extraction of background and then check which ones of those 'detected' flares lies above the 3σ level of the background. Only those will be contribute to the 'real' flares.
- We started by guessing an initial start and end time. We remove the 'detected' flares from the light curve (single day) based on the initial 4 increasing 3 decreasing points. This marked as our initial guessed start times (first point of the increasing trend) and end times (last point of the decreasing trend) respectively. We removed all the flares between the initial and final start times connected the each flare. After removing all the flares, the rest of the data is assumed to be contributing to the 'initial guessed' background.
- The intensities of definite flares on certain days were equal to those of noise and even the background on more active days. This made it difficult to find a straightforward algorithm for flare detection using a background, which made us implement different background models.

3.0.1 Constant Background fits

- We fitted a constant line to the full day data after removal of the initial detected flares. Those flare peaks whose count rates were 3σ times higher than the background level were eliminated, and the rest of the data was assumed to be contributing most to the background.
- We repeated this process till the background level i.e. $const + 3\sigma$ level converges. Then taking this as the background level, the 'start times' and the 'end times' of the flares were assumed to be starting above or at this level. A condition was put forth that the end times of a flare must occur before or at the start time of the next detected flare.

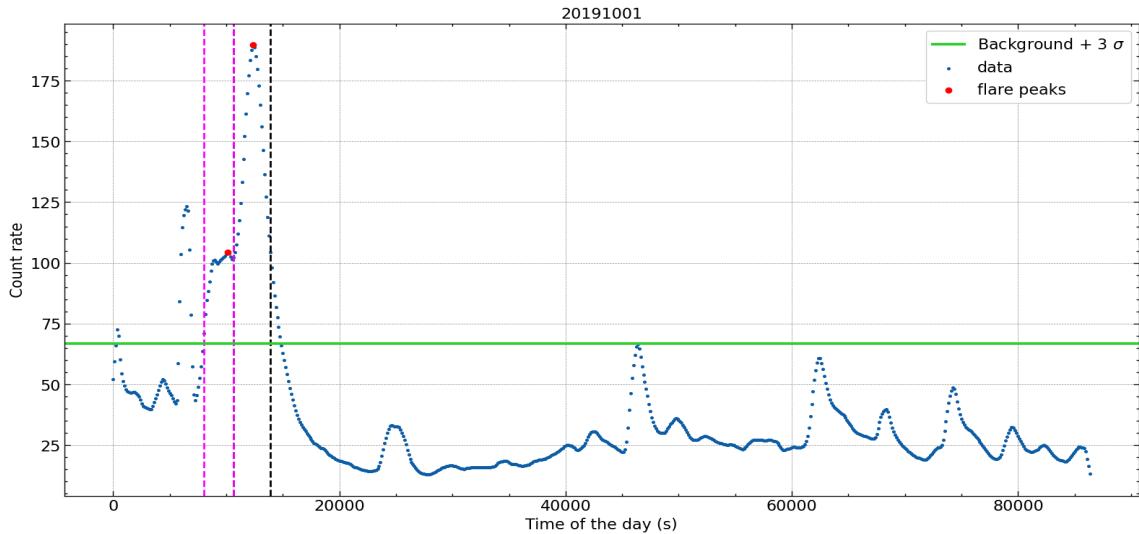


Figure 3.4: Light Curve with cleaned detected flares with end times in black, start times in magenta, and a constant background level in green colour (01 Oct, 2019)

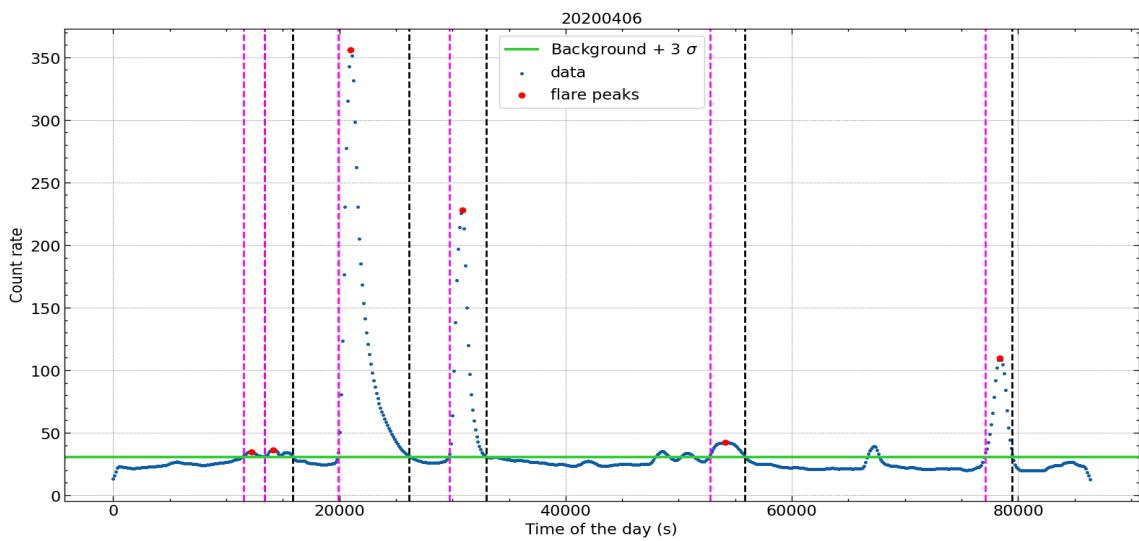


Figure 3.5: Light Curve with cleaned detected flares with end times in black, start times in magenta, and a constant background level in green colour (06 Apr, 2020)

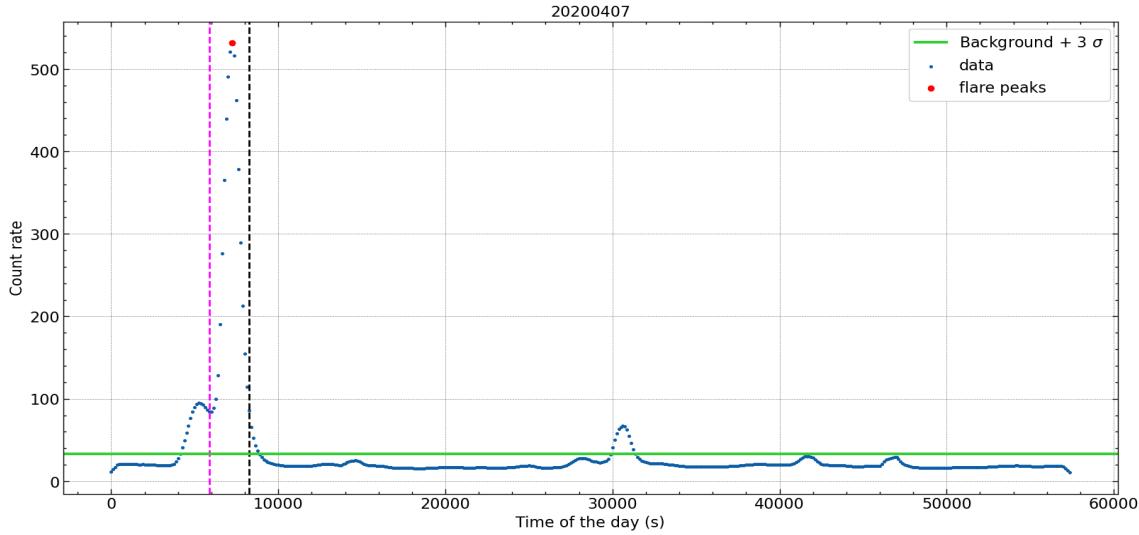


Figure 3.6: Light Curve with cleaned detected flares with end times in black, start times in magenta, and a constant background level in green colour (07 Apr, 2020)

- After this we get the start times, end times, and the peak of the flare. Using these parameters, we tried to fit a convolution function i.e. A gaussian with and an exponential decay to our flares.

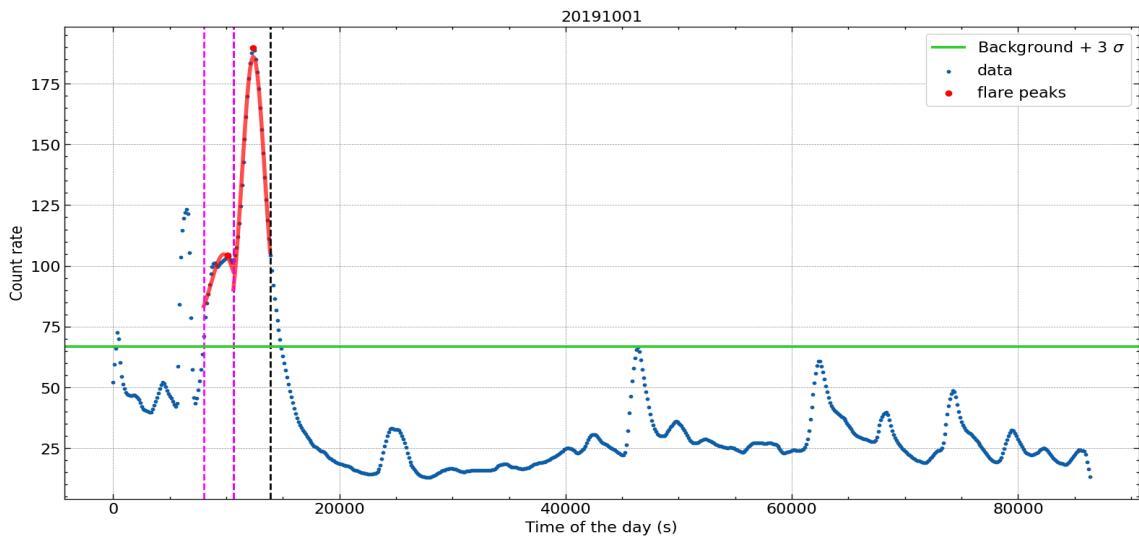


Figure 3.7: Light Curve with fitted flares (red curve) with start times in magenta, end times in black, and a constant background level in green colour (01 Oct, 2019)

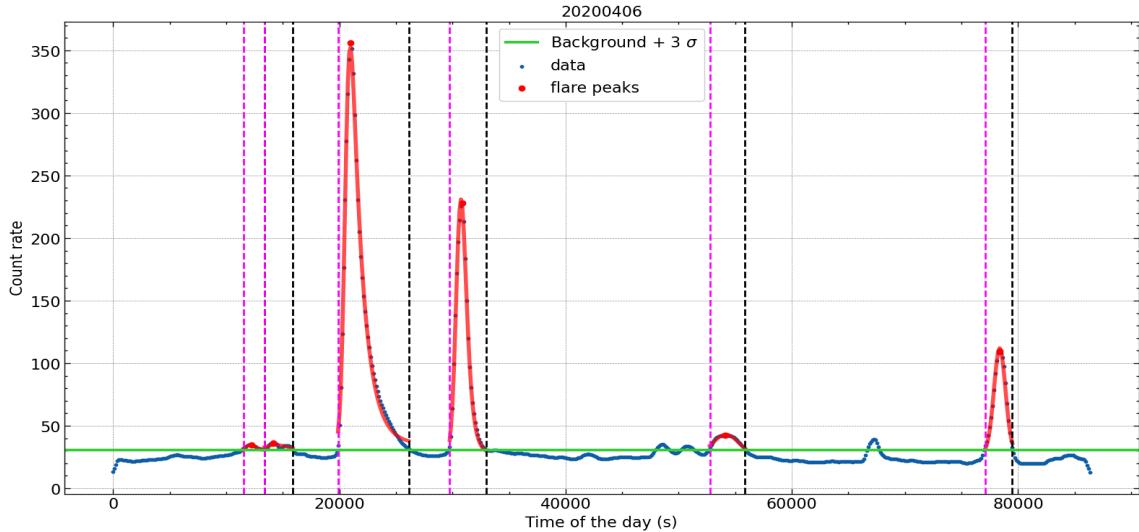


Figure 3.8: Light Curve with fitted flares (red curve) with start times in magenta, end times in black, and a constant background level in green colour (06 Apr, 2020)

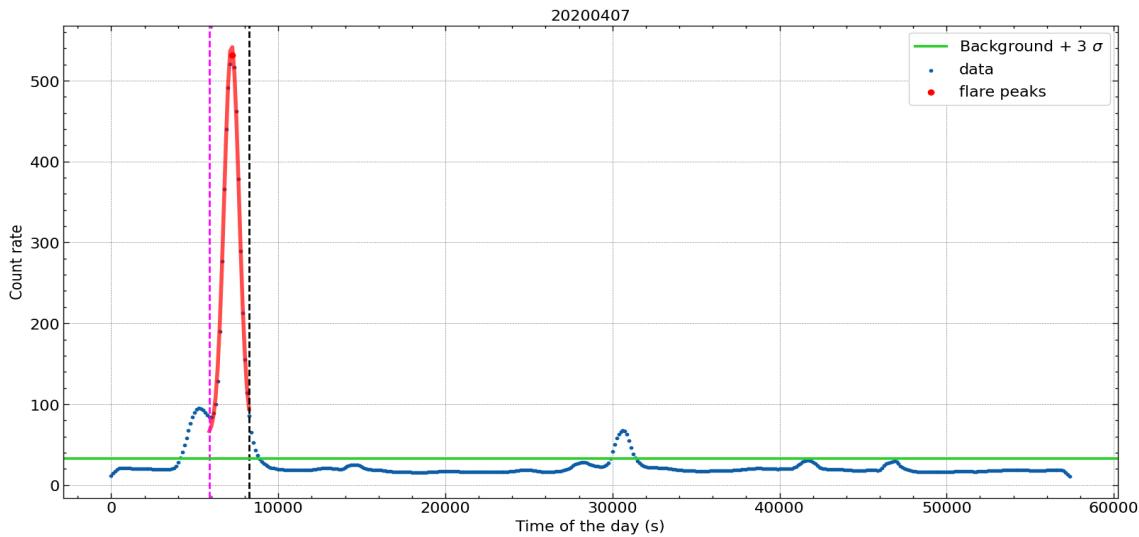


Figure 3.9: Light Curve with fitted flares (red curve) with start times in magenta, end times in black, and a constant background level in green colour (07 Apr, 2020)

3.0.2 Linear and Polynomial Background fits

- We fitted a straight line to the noise and remove outliers. i.e anything that goes more than $2.5 - 3\sigma$ away from the line. This is done, because in the case of multiple flares, some residual flare data remains from the minima between peaks.
- For the remaining data, we fitted a polynomial to it, or a set of $3 - 4$ straight lines. We used this fitted background for the background subtraction from our data.

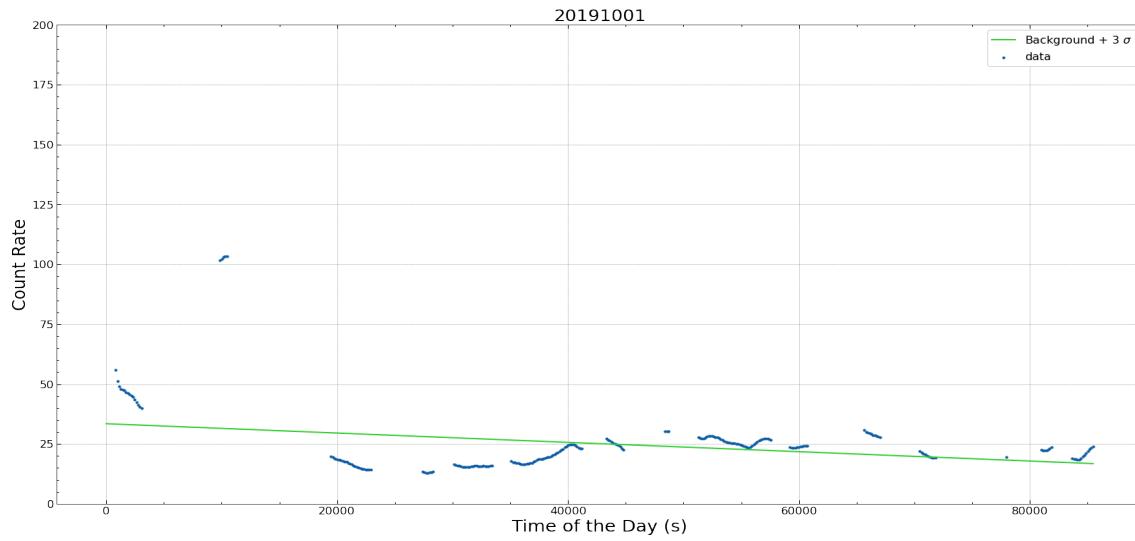


Figure 3.10: Cleaned data (flares removed) with linear background fit in green colour (01 Oct, 2019)

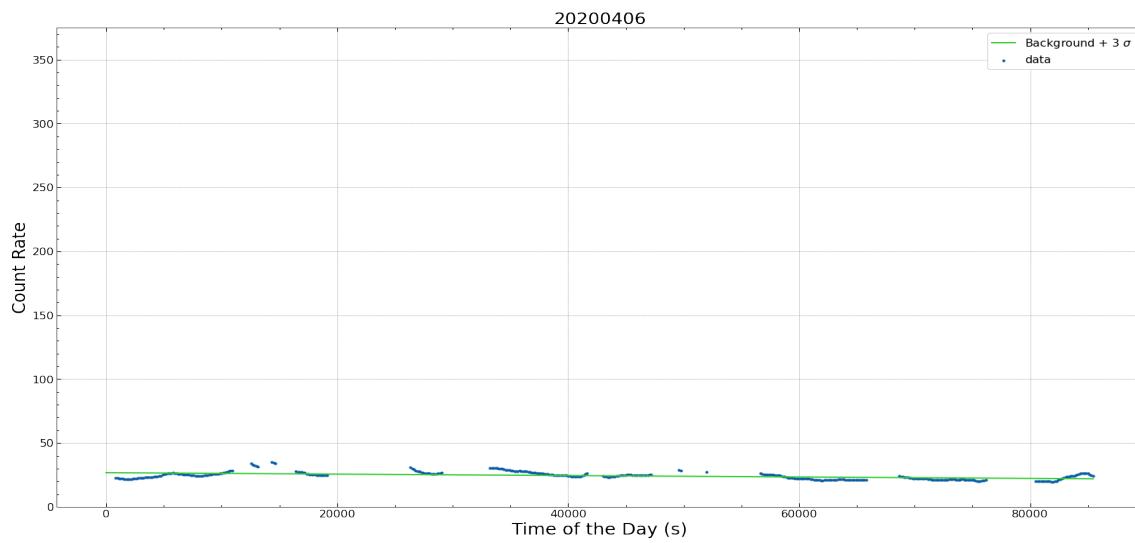


Figure 3.11: Cleaned data (flares removed) with linear background fit in green colour (06 Apr, 2020)

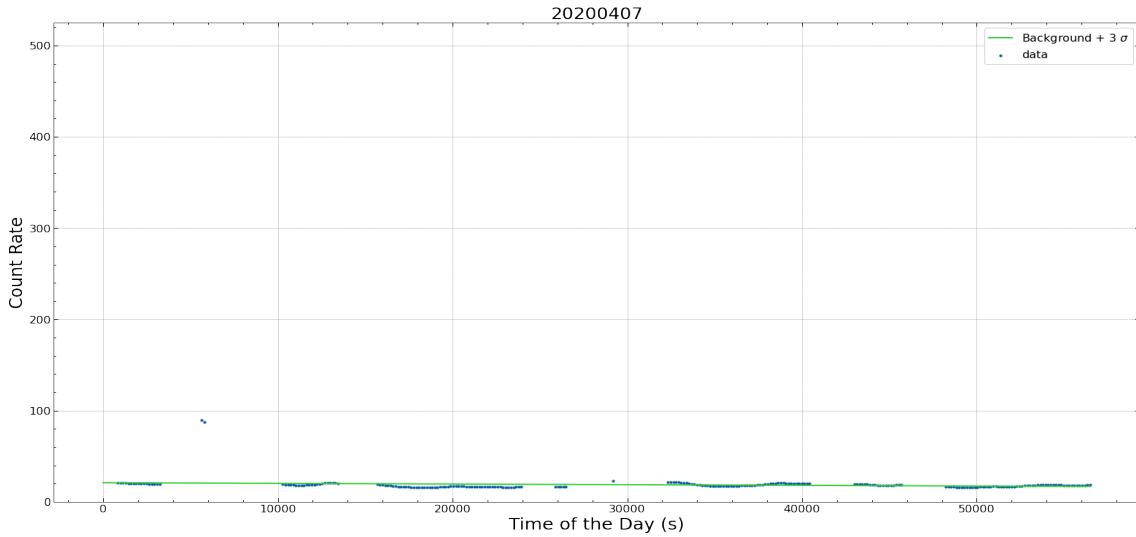


Figure 3.12: Cleaned data (flares removed) with linear background fit in green colour (07 Apr, 2020)

- After background subtraction, we found the start and end times. Based on the initial algorithm, we found the start times using increasing consecutive points method. The end times are found by passing a line which has the slope of the background at that point, through the point. When this line intersects the light curve again, we marked the point as the end time. Flares which were too short(< 20 mins) are rejected.
- For the remaining flares, curve fitting is done. This curve is the convolution of a gaussian rise and an exponential decay. [Gryciuk et al. \(2017\)](#).

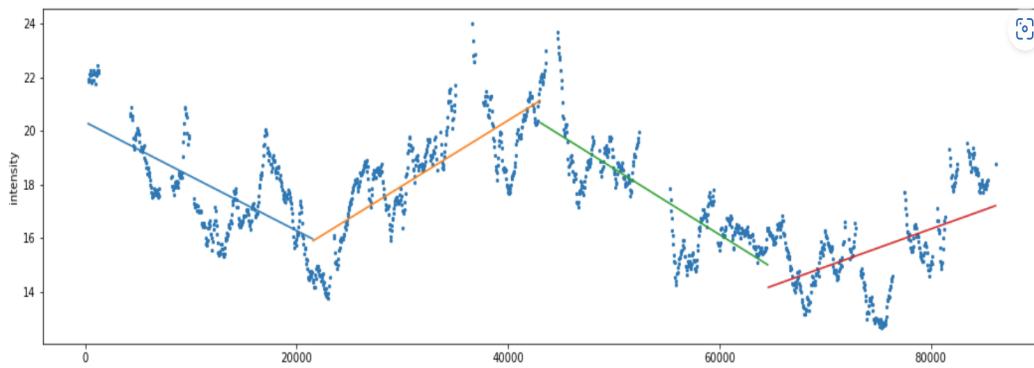
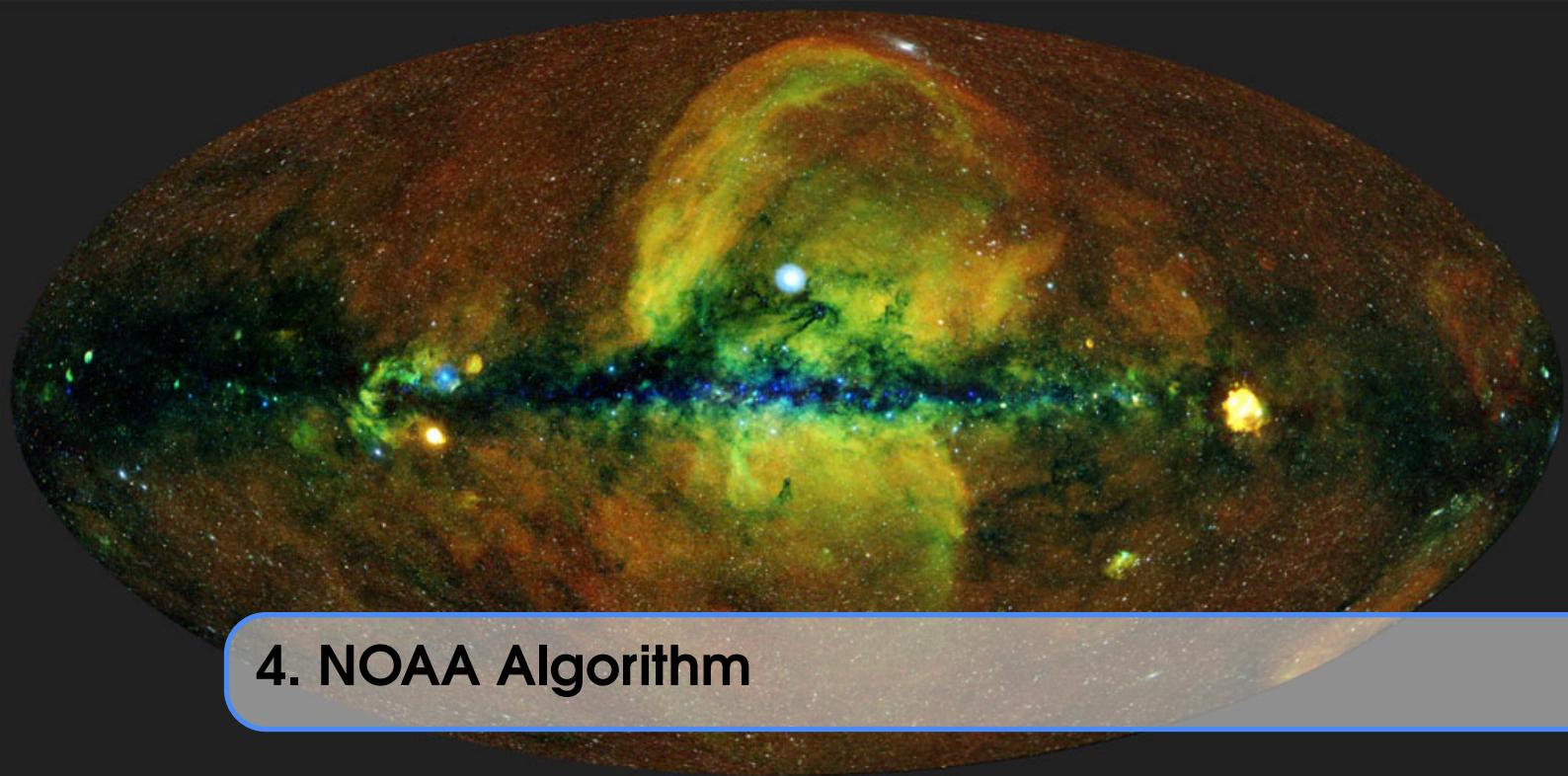


Figure 3.13: Scatter plot of background and multi-line fit



4. NOAA Algorithm

- The NOAA detection algorithm was followed in order to automatically detect the start time, peak and end time of the flares. The NOAA detection algorithm implemented by me proceeds in the following way:
 - All such instances of data that contain 4 consecutive increasing points with the last point being 1.4 times greater than the fourth were found (Note: The value of the factor is flexible depending on the dataset.)
 - For each such instance, we temporarily assigned it as the start time of the flare and detect the nearest peak and assign it as a peak. Now find and assign the end of the flare to be where it drops to half the value of the peak (0.5 times (peak value + start value)).
 - In the rising part of a particular flare we might end up finding multiple instances of start times for a particular peak which will also result in multiple end times. Hence, the earliest start time in such cases were chosen and then assign the corresponding end time using the identified peak.
 - This algorithm might end up finding small peaks in data which may not always represent the biggest peak between the start and end time as it detects the first local maxima after 4 consecutive increases. Hence while searching for an end time, if we find a higher peak before we find the value at which the flare ends, we update the peak value and continue to search for the end time of the flare with the new peak value in consideration.
 - Finally, for flares that don't reach an ending within a given day, we assign the end time as the last datapoint in the graph.
- Flares were removed (data points between the start and end time of the flares) from the overall data and assign the remaining data points as the

background radiation for the day.

- Mean and Standard Deviation of the background was calculated, and all the data in the background which is more than ± 1 std around the mean were eliminated and the process was repeated until the std is down to a certain threshold, the background at this point can be safely considered as the true background.
- Various fit functions were explored like the one mentioned in the Flare Characteristics from X-ray Light Curves paper: Convolution of a Normal distribution function and an Exponential decay function and a convolution between a normal distribution and an inverse power law function. Parameter tweaking in the functions according to a given elementary flare in consideration. This is done in order to identify whether the detected flare is indeed a valid flare and what mathematical function best describes a solar flare.

5. Local Extrema Algorithm

This method is inspired by the section 2.2 of [Aschwanden and Freeland \(2012\)](#) but the method described in this section has significant deviation from it in many cases.

Firstly, we need to do binning with a certain bin size, and different bin sizes were seen to be suitable for different datasets but mainly it will vary within 50s to 300s. For the example mentioned in this paper, I have used `binsize=50s`. We have used mean for binning. It was followed by removal of data gaps by employing the `dropna()` function from the pandas library.

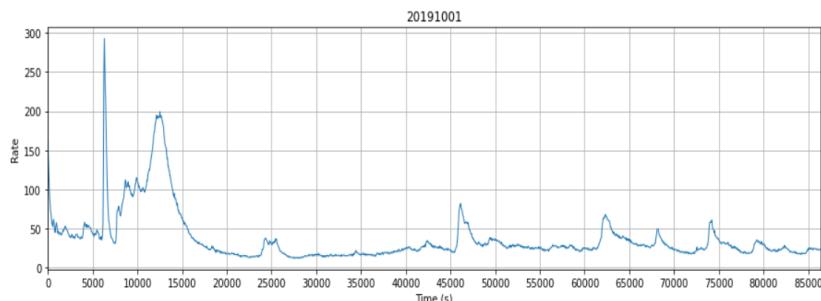


Figure 5.1: Signal after Binning

Secondly, data smoothening is done with boxcar averaging method with a window size, also dependable on the data set (mainly within 10 to 25). In this case, I have used `window size=15`. Before this step, we can include a Noise Elimination Step, where we will eliminate data points outside the $\text{mean} \pm \text{threshold}$ limit, and doing this iteration for at least 5-6 trials.

Then, we will find 'local' minimas and maximas to a certain extent of locality, i.e., within how many neighbouring points, the extrema is detected. I have set order of locality to $n=10$ here.

For finding threshold, those maximas were removed where the slope of the

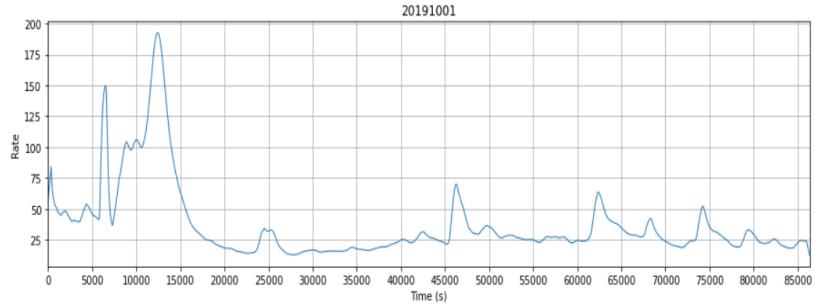


Figure 5.2: After smoothening in Boxcar Average Method

increasing part is less than a certain threshold, which here is set to 0.2. This may change in different cases. Next, we search for those maxima which are between two minima points, as they may be potential flare peaks. This is just a heuristic way to finding potential flares, so that we can exclude them to get the background of the signal. After removing those parts between consecutive minimas, which contain one or more maxima in between, we take median of the rest of the signal (lets name the flareless data as noisy background) to get the background. One can take mean also and see which one suits better. We assumed here that background is a constant function through out the whole day, and have drawn a straight line as background line, while we can also do linear fitting or any other function, and they also may suit better than constant function, but constant function has worked quite well in the given case.

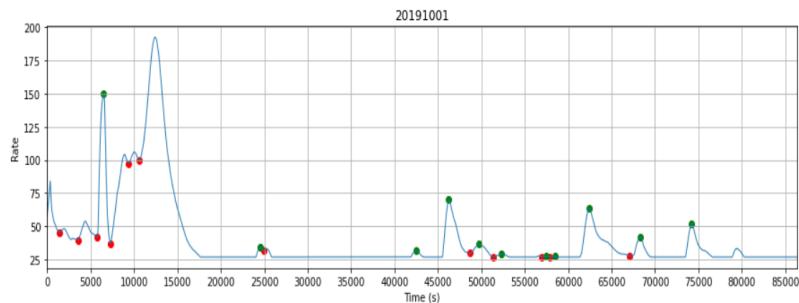


Figure 5.3: Setting Background Constant with marked Peaks in Green points

We take the standard deviation (σ) of noisy background (defined before). Then we eliminate those peaks which are below the level of background $\times \sigma$. We replace those minimas with background value we determined before, which are below the background line.

We get a much nicer signal now, which have a very steady nice-looking background, with peaks on it, smooth and nice. Where ever the shapes start rising from the background, we mark the starting point, and where ever it reaches the background, we mark the end point. (It is different from the idea used in the papers, but the background detection algorithm we used here, gives a background a bit higher than that one from the paper, so this makes sense). As background, we have taken a window of length 1 here, which contains the background.

Another approach can be taken regarding detecting the background as follows: The background flux is defined from the median flux in a time interval ($t_s - t_{min}, t_s$). However, the problem with this method is that the background is different

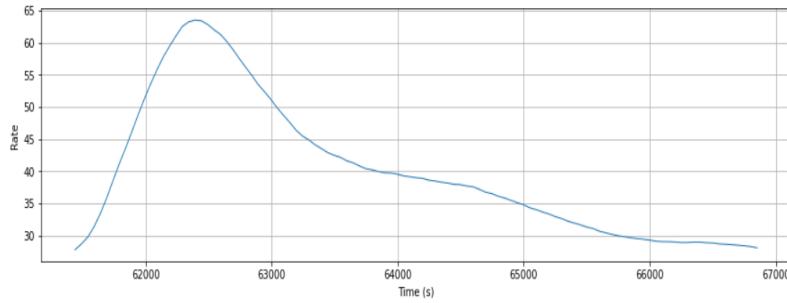


Figure 5.4: A single flare

for every local minima, and it is not constant for the complete dataset, as shown in Figure 5.5. We propose a way to make it constant. We first detected the background fluxes for every local minima point and took the median of all these fluxes to get the global background flux as shown in Figure 5.6.

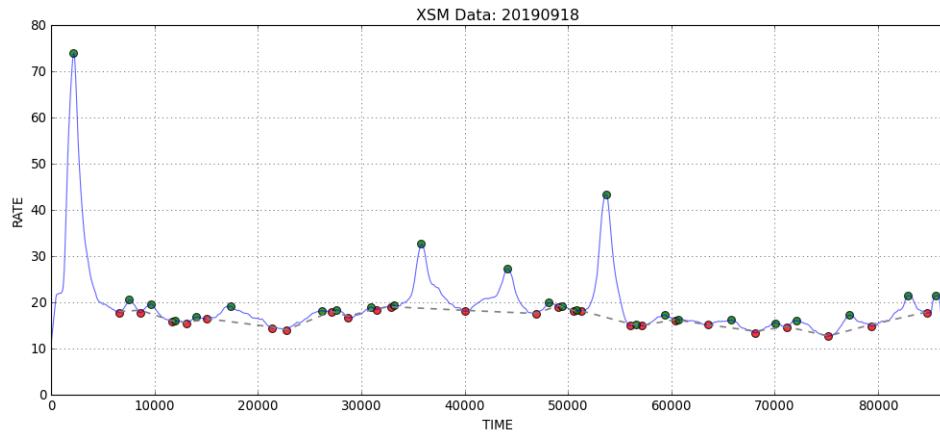


Figure 5.5: Figure showing the local maxima with green spots, local minima with red spots. The order chosen is 10. The grey-coloured dashed line shows the background flux.

We then can look into a particular flare in the signal and can fit its shape with the function Elementary Flare Peak described in the paper of Gryciuk et al and get the parameters needed to describe a flare event. We can see how efficient our algorithm was from its shape similarity to the ideal flare of EFP itself.

Before the background determination method and all steps afterwards, we can also apply Chebyshev fit on the obtained flares as below.

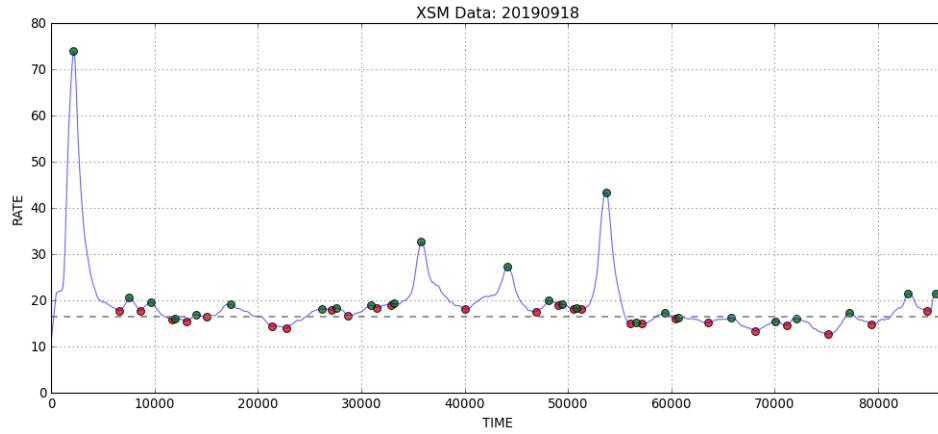


Figure 5.6: Figure showing the local maxima with green spots, local minima with red spots. The order chosen is 10. The grey-coloured dashed line shows the constant background flux.

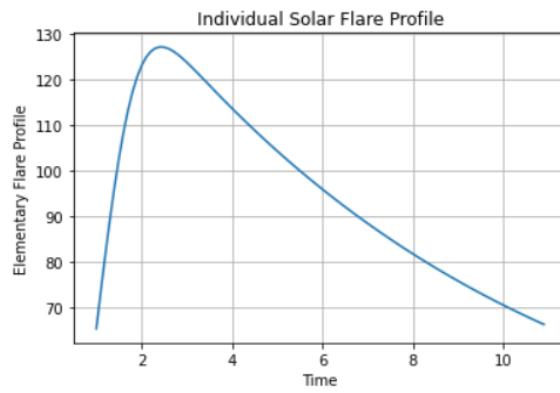


Figure 5.7: Ideal shape of Solar Flare according to EFP

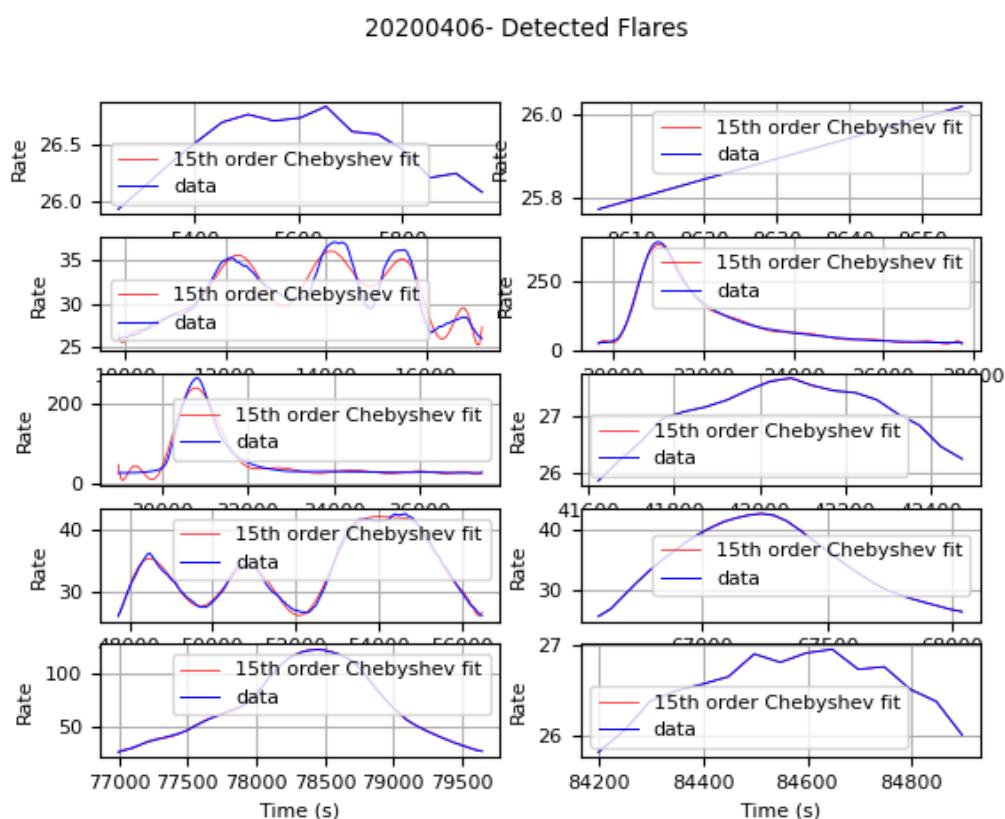
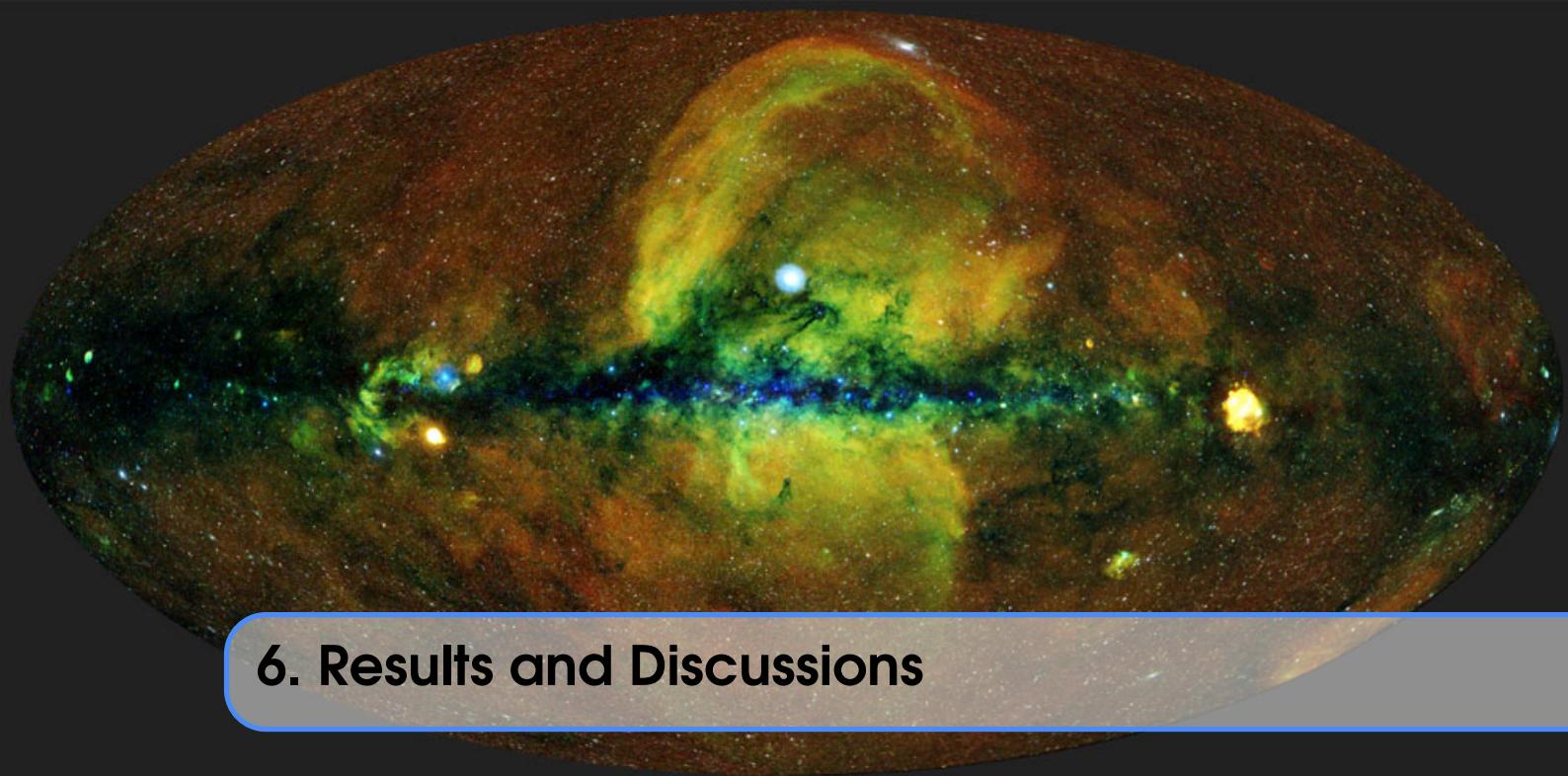


Figure 5.8: Method 5- Identified flares and Chebyshev polynomials fit



6. Results and Discussions

Throughout the analysis, optimizing bin sizes and flare duration proved to be a challenge. Smoothing also caused loss of resolution in the data. It lead to the detection of background noise as potential flares and skipping of possible flares, which were marked by manually eyeballing the data.

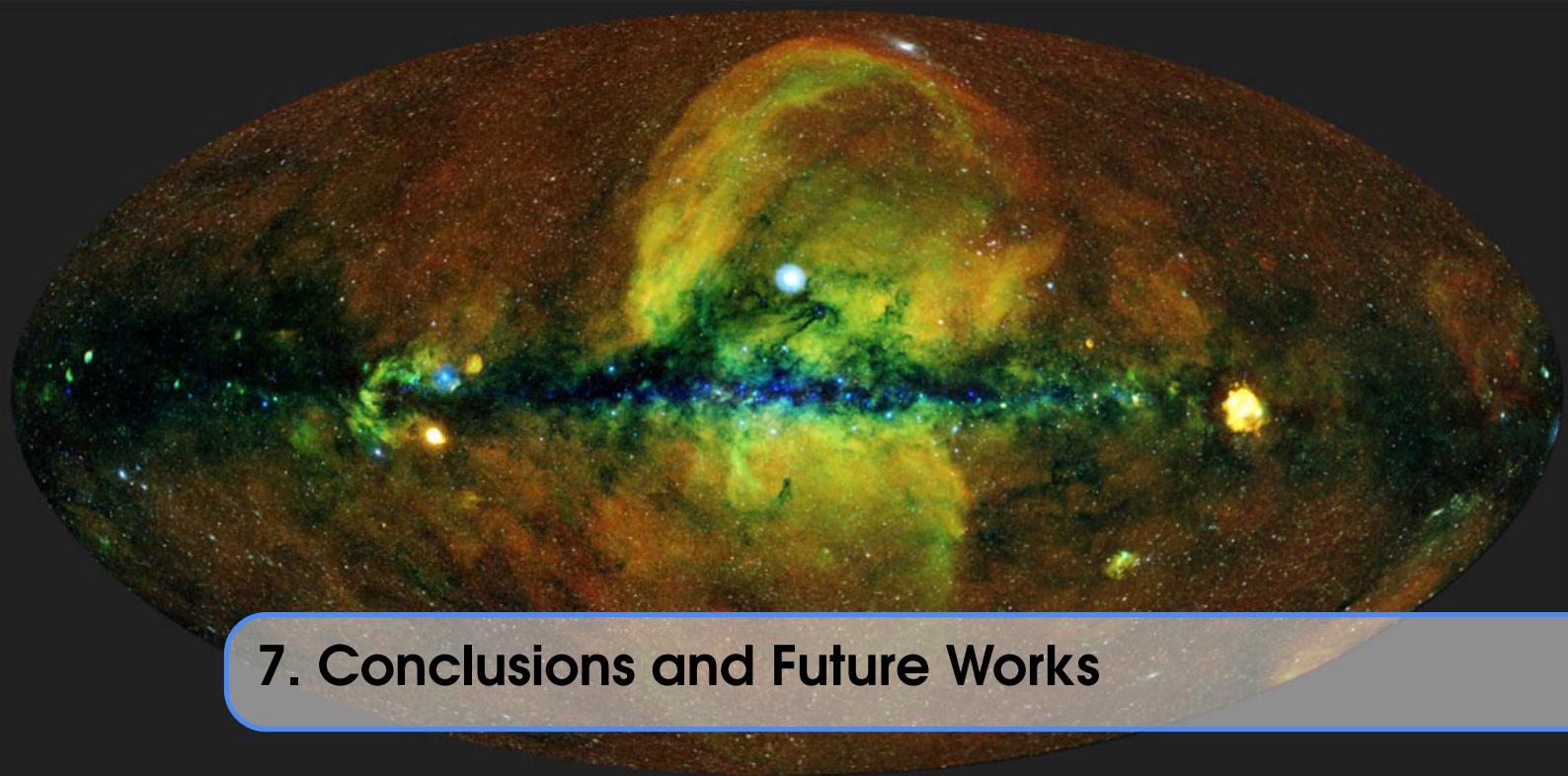
The increment in the threshold value for the slope in the trigger algorithm caused various peaks to be non-detectable. We have to find a better way to resolve and fix a particular threshold. Inaccuracies in the peak detection algorithm cause these flares to not be removed and trickle down into the background. This was the reason we had to include the removal of data beyond 3σ from the background. Moreover, the background at that point had ended up containing many flares. Often, the background data from the algorithm did not span the whole day and this lead to over-fitting. This made it difficult to optimize a single background fitting technique for different days. Apart from this, this algorithm would often capture multiple flares and call them a single flare. This led to bad fits that were difficult to resolve.

The NOAA algorithm also missed detection of flares instances containing double peaks as we only detect global maximums between the start and end times. Hence, a formulated procedure to assign consecutive local minima as start and end times in such cases which is to be implemented.

Also, the Local Extrema Algorithm identifies several sets of three points as potential flares that can't fit according to the elementary flare criterion. Furthermore, to reduce the time complexity, the algorithm wasn't repeated for identifying more minor flares.

For the datasets in the year of 2019, we had very good data as we had solar minima that time and it marked start of a new solar cycle. For later times, many of the datasets could not be handled well with this algorithm(Chirag).

The algorithm(Gourav) is unable to detect low intensity flares



7. Conclusions and Future Works

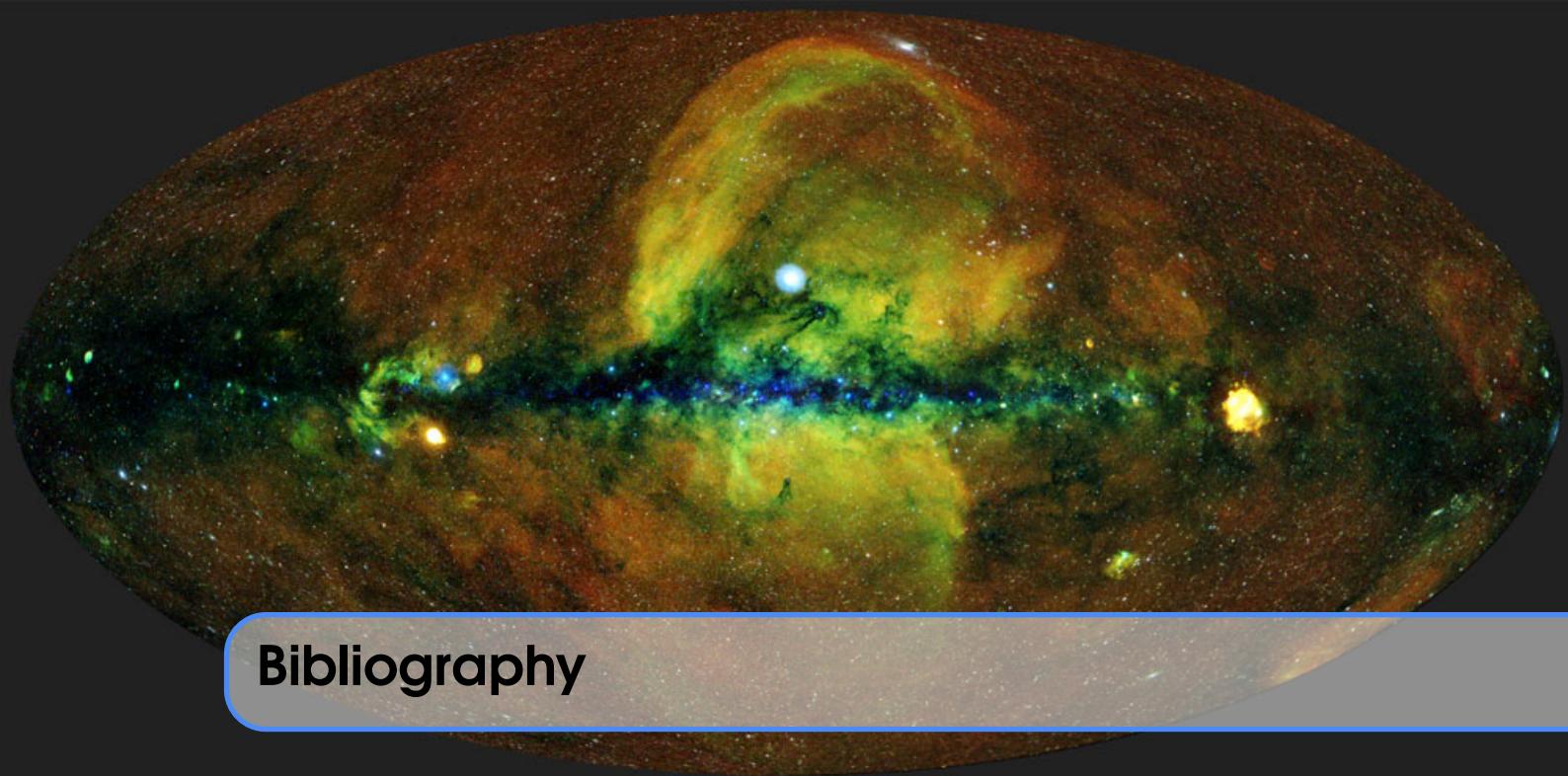
Based on the count rates and the Flares detection algorithms used, we learnt that the solar background fluctuates slowly over time and the solar minima occurred in the year 2020. We found that not all flares have a gaussian rise and an exponential decay. Even if the detection algorithm seems to be identifying them as flares, the convolution function was not completely accurate. Few flares were better fitted with gaussian only fits and few with just decaying exponential fits. We figured out that Fast Rise and Exponential Decay (FRED) could be a potential fit to our detected flares. We hope to implement this model in future which will also help determine different Flare parameters.

The methods used are not yet robust against semi-empty or empty datasets which occurred during eclipses or days where data was not collected by Chandrayan-2. Assigning a threshold for the number of points in the data set for it to be under consideration and making the bin size dynamic for a variable number of data points are possible solutions. We intend to develop better techniques to differentiate between flaring and background regions and to identify multi flares accurately.



8. Acknowledgements

We acknowledge the use of data from the Solar X-ray Monitor (XSM) on board the Chandrayaan-2 mission of the Indian Space Research Organisation (ISRO), archived at the Indian Space Science Data Centre (ISSDC). XSM was developed by Physical Research Laboratory (PRL) with support from various ISRO centers. We also thank Indian Institute of Technology (IIT), Bombay Astronomy CLub Krittika for giving us the opportunity to work in collaboration on the 'Stellar Flares' project.



Bibliography

- L. V. Mirzoyan. Instability and evolution of stars. *Izvestiya Akademii Nauk Armyanskoi*, 16:126–133, January 1981.
- E. Parsamian and E. Chavira. New Flare Stars in the Pleiades. *Boletin de los Observatorios Tonantzintla y Tacubaya*, 5:35–40, May 1969.
- H. M. Tovmassian, V. P. Zalinian, N. A. Silant'ev, O. Cardona, and M. Chavez. The unified phenomenological model of light curves of stellar flares. *A&A*, 399: 647–651, February 2003. doi: 10.1051/0004-6361:20021754.
- M. Gryciuk, M. Siarkowski, J. Sylwester, S. Gburek, P. Podgorski, A. Kepa, B. Sylwester, and T. Mrozek. Flare Characteristics from X-ray Light Curves. *Sol. Phys.*, 292(6):77, June 2017. doi: 10.1007/s11207-017-1101-8.
- N. P. S. Mithun, Santosh V. Vadawale, Aveek Sarkar, M. Shanmugam, Arpit R. Patel, Biswajit Mondal, Bhuwan Joshi, P. Janardhan, Hiteshkumar L. Adalja, Shiv Kumar Goyal, Tinkal Ladiya, Neeraj Kumar Tiwari, Nishant Singh, Sushil Kumar, Manoj K. Tiwari, M. H. Modi, and Anil Bhardwaj. Solar X-Ray Monitor on Board the Chandrayaan-2 Orbiter: In-Flight Performance and Science Prospects. *Sol. Phys.*, 295(10):139, October 2020. doi: 10.1007/s11207-020-01712-1.
- Markus J. Aschwanden and Samuel L. Freeland. Automated Solar Flare Statistics in Soft X-Rays over 37 Years of GOES Observations: The Invariance of Self-organized Criticality during Three Solar Cycles. *ApJ*, 754(2):112, August 2012. doi: 10.1088/0004-637X/754/2/112.
- Thomas P. Robitaille, Erik J. Tollerud, Perry Greenfield, Michael Droettboom, Erik Bray, Tom Aldcroft, Matt Davis, and Adam et. al. Ginsburg. Astropy: A community Python package for astronomy. *A&A*, 558:A33, October 2013. doi: 10.1051/0004-6361/201322068.