ExoCoDe: MODELING TRANSITS EVENTS VIA STATISTICAL AND MACHINE LEARNING TOOLS

by Karakuts Denys

A Capstone Project

Presented in Partial Fulfillment of the Requirements for the Degree

Master

American University Kyiv

2024

APPROVED BY:

Jacek Leśkow

Ph.D., Dean of AUK EPAM School of Digital Technologies

ORCID: 0000-0003-2228-393X

Abstract. *This capstone project uses statistical and machine learning algorithms to detect exocomet transits in TESS telescope data. Exocomets, distinguished by their unique, asymmetric light curves, present a detection challenge due to their subtle signatures compared to planetary transits and light intensity. We develop a framework that integrates data preprocessing, feature extraction, visualizations, statistical methods, and machine learning regressors to characterize these transits efficiently. The project is built upon the existing progress of astrophysical research. It aims to enhance our understanding of exocometary activity, uncovering the potential of machine learning and statistical analysis in astronomical data interpretation.*

Keywords: *exocomet, anomaly detection, time-series, machine learning, star brightness.*

Table of Content

Introduction

The Transiting Exoplanet Survey Satellite (TESS) is a NASA astrophysics mission that aims to discover thousands of exoplanets around bright stars nearby. Since April 2018 TESS has been using several wide-field cameras to capture the brightness of stars over large areas of the sky, looking for temporary dips in brightness caused by planetary (or cometary) transits. As the primary mission, this method allows TESS to detect exoplanets as they pass in front of their host stars and temporarily block a portion of the starlight. Even though looking for exocomets is not the main goal of the TESS mission, it could also be done by analyzing data from this telescope in a similar manner as exoplanets, however, this task is more challenging. The group of Ukrainian astrophysicists, led by Yakiv Pavlenko, the Chief Research Fellow of the Main Astronomical Observatory NAS of Ukraine are conducting the research to find evidence for the presence of exocomet events in TESS data [1].

Exocomets are remnants of the material from which planetary systems are formed. By studying them, scientists can gain insights into the processes of planetary formation and the early history of planetary systems. Furthermore, exocomets may have played a role in delivering water and organic compounds to planets, including Earth, which are essential ingredients for life [2]. Understanding their role in the context of other star systems could shed light on the potential for life elsewhere.

The data collected by the TESS telescope is translated into time series, recording the brightness of stars over time [3]. This characteristic makes the dataset particularly suitable for analysis using statistical and machine learning techniques, which excel at identifying patterns and anomalies in sequential data. Statistical methods can be employed to analyze the periodicity, trends, and noise in the light curves, providing insights into the underlying astrophysical processes. Machine learning algorithms, especially those designed for time series analysis, can be trained to detect specific

signatures, such as those indicative of exoplanet transits or exocomet activity. By leveraging these advanced analytical tools, researchers can more effectively sift through the vast amounts of TESS data to extract more sense from it.

The software system has been named ExoCoDe, an acronym that represents 'ExoComet Detection system'.

The first chapter of this capstone project focuses on the architecture for the system solving the task of the project. It begins by outlining the requirements, both functional, non-functional, and constraints. The technologies section reviews the programming language, frameworks, visualization tools, and machine learning techniques used. The architecture subsection documents the general system design, including system context and container diagrams, as well as the database structure.

The focus of the second chapter shifts to the description of the data from the TESS Telescope and the analytical mechanics applied for it. The chapter breaks down the data manipulation pipeline algorithms, including the processes for normalizing, trending the data, as well as the methodology for identifying moving difference data. It explains how the project determines appropriate windows for analysis of potential transit events.

In the final third chapter, the project presents the analysis and visualization of exocomet events. It provides the techniques used to pinpoint ingress and egress points of transit events with machine learning. The chapter also describes the web interface developed to facilitate data exploration, along with specific sections dedicated to sectors and stars, providing a comprehensive view of the project's outcomes.

Literature Review

Exploration of other stellar systems is a vast field of research. The transit method, used for exploring the inhabitants of these systems, is currently the most effective, resulting in the discovery of 4,151 planets, according to NASA data [4]. This method can also be applied to smaller objects orbiting stars, including comets. The primary sources of data for the transit method are two telescopes: Kepler [5] and TESS [6]. This capstone project will focus on the latter.

The primary inspiration for this project is the previously mentioned research by Ukrainian scientists from the Main Astronomical Observatory [1]. They have taken into consideration the star Beta Pictoris, which is already known to host exocomet activity [7] and are investigating it for additional data sectors. However, their research is conducted on an ad-hoc basis, focusing on a single star.

There is a very limited amount of research on the topic of automated detection of exocomet activity. This is due to the lower interest from the scientific community compared to planetary transits, with additional challenges posed by the high volatility of the data and the less pronounced significance of exocomet patterns. The most prominent research is the work of Kennedy et al. [8], which defines the theoretical foundations for identifying cometary activities for several stars observed by the Kepler telescope. Furthermore, there are a few papers dedicated to manual, ad-hoc exocomet detection, such as that by Rappaport et al. [9], which also propose methodologies for finding and characterizing exocomet activities.

One of the most widely used and cited algorithms for processing time-series data of star fluxes is the research conducted by Michael Hippke et al. [10]. This algorithm is instrumental in removing noise from the data, thereby revealing patterns that could be further utilized for the detection of exoplanets and exocomets.

Chapter 1. Software architecture analysis

1.1. Requirements

1.1.1. Functional requirements

Extracting Data and Processing:

- The software must be capable of inputting the data from the TESS (Transiting Exoplanet Survey Satellite) telescope and complimentary sources of data.
- It should preprocess this data to a format suitable for analysis, including normalization.

Visualization Tools:

- Develop a user-friendly interface for visualizing the TESS data and the results of statistical and machine learning analysis.
- Include interactive elements such as zoom, filter, and highlight features for detailed examination of the data.

Statistical Analysis Tools:

- Implement various statistical methods, like time-series analysis, moving averages, comparative analysis to process signals of different stars and identify potential exocomet transits in the data via parametrizing these transit events.
- Integrate machine learning algorithms capable of characterizing exocomet transit events.

1.1.2. Non-functional requirements

Performance:

- The system should be able to process large volumes of TESS data efficiently.
- The system should ensure quick response times for user interactions and data processing tasks.

Usability:

- Even though the system will have a limited number of users, its user interface should be intuitive and user-friendly.
- Provide clear instructions to assist users in navigating and utilizing the system.

1.1.3. Constraints

Limited budget:

- Given the project's narrow focus and limited budget constraints, the software must be cost-effective, using primarily open-source technologies due to the lack of substantial funding. This approach limits the choice of technologies to those that are more affordable.

1.2. Technologies

1.2.1. Language

The main language of the project will be **Python**, as it provides possibilities for scientific research, and specifically in the field of astrophysics exploration. Also, machine learning algorithms are covered excessively with this domain language.

Also, the pre-processing modules will be given in the form of **Jupyter Notebook** [11] files, as it is one of the main instruments of current scientific research, which provides possibilities for the scientists to fine-tune and optimize the execution of the data-preprocessing steps, and our astrophysicists are very well familiar with a tool.

1.2.2. Visualization

In order to display the outcomes of the project visually, taking into account the specific needs and limitations of the project, the **Dash** library [12] in Python has been selected as an appropriate option for multiple reasons:

- Dash is an open-source library, which aligns well with the project's constraint of a limited budget. There's no need for expensive licenses or subscriptions, making it a cost-effective solution for visualization needs.
- Dash enables the creation of interactive, web-based data visualization interfaces with relatively simple Python code. This ease of use facilitates rapid development, allowing us to focus more on the core functionalities of the project.

- Dash provides extensive support for customizable and interactive visualizations in a user-friendly manner. It supports various types of plots and graphs that can be interactive, enhancing the user experience, providing filters, zooms, dropdowns, etc.

- Dash applications can be scaled to handle large datasets, which is crucial given the potentially extensive TESS telescope data. Its performance remains consistent even as the dataset grows.

1.2.3. Machine Learning

For the lightweighted machine learning tasks it was decided to use **Scikit-learn (sklearn)** library in Python [13], as it's also open-source, highly maintained, and offers a comprehensive suite of well-established machine learning algorithms for classification, regression, clustering. The library is known for its user-friendly interface and consistent API, which simplifies the process of implementing and testing different machine learning models.

While it is designed with ease of use in mind, it also offers sufficient performance for many machine learning tasks. Given that the project is research-oriented, sklearn's extensive selection of algorithms and preprocessing methods makes it a valuable tool for experimentation and discovering the best approaches.

1.3. Architecture

To describe the structure of the project let's use the C4 model [14]. It will allow us to look at the system design from different angles and scales. For the purposes of the project, it was decided to draw the first two levels of the C4 model, which are System Context and Container diagrams.

1.3.1. System Context diagram

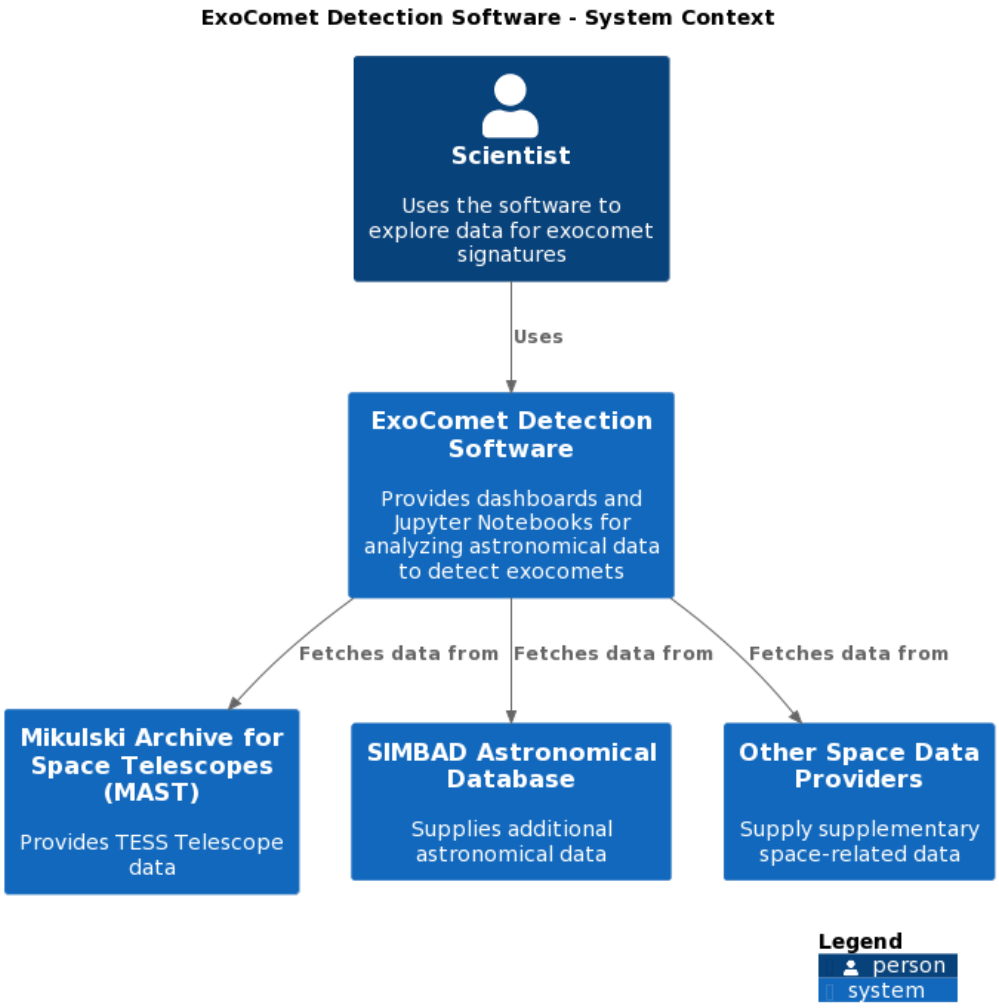


Fig 1. System Context Diagram for ExoComet Detection software [15].

Firstly, we need to understand the system surroundings, which actors will be using the system, which external systems are used and in which manner. To do this, we

can exploit the System Context diagram of C4 model, where the system in question should be drawn with no details, to see the bigger picture around.

The main person of the system usage will be a scientist, who is using the system to explore and manipulate with the telescope data in a more suitable and comprehensible way.

Two main systems, which are used to get the data, are 1) Mikulski Archive for Space Telescopes (MAST) [16], to retrieve the flux files, which are the main resource for the exploration, and 2) SIMBAD Astronomical Database - CDS (Strasbourg) [17], which provides the info about stars' characteristics.

1.3.2. Container diagram

Going deeper into the internal structure of the system, we will use the Container diagram from C4 Model, in order to show building blocks of the system.

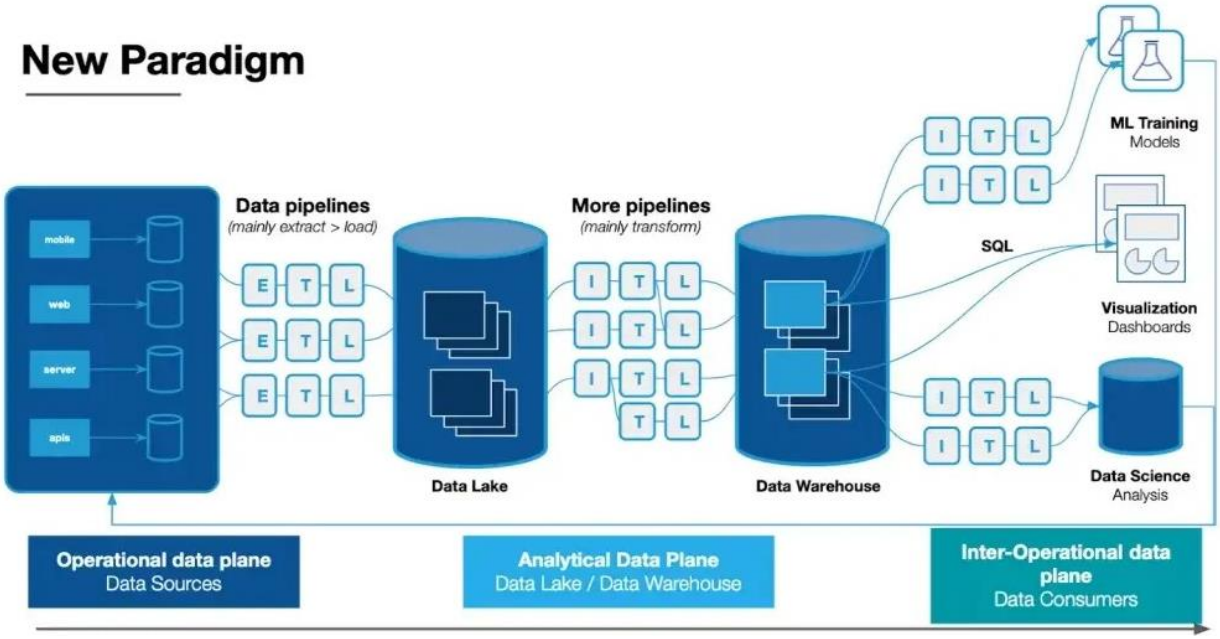


Fig 2. Data Pipeline architecture [18].

To start with, we need to select the architectural style we want to use for the system. Based on the requirements for the system, it was decided to use Data Pipeline architecture, which provides the most suitable structure to solve the problem in place, as we have huge amount of data which should be processed efficiently [18].

For the project we've defined the following containers:

Data Extracting Module: This module is responsible for fetching and aggregating astronomical data from external sources like the Mikulski Archive for Space Telescopes (MAST) and the SIMBAD Astronomical Database. It acts as the initial data ingress point.

Transformation Module: This module processes the raw astronomical data to make it suitable for analysis. It systematizes the data, finds the trends, calculates relevant statistics, filters out unnecessary or irrelevant information, and focuses on data that is most likely to indicate the presence of exocomets.

Windowing Module: The windowing module slices the processed data into smaller, manageable segments or 'windows'. This is crucial for handling large datasets and for calculating statistical measures that could indicate exocomet activity within these specific data segments.

Machine Learning Module: This module employs machine learning algorithms of logistic regression to analyze the windowed data. It focuses on improving the results of characterizing the potential exocomet transit events in a star system.

Visualization Module: This module is essential for interpreting the analysis results. It provides scientists with visualization tools to investigate processed and analyzed data. This module helps in visualizing patterns, trends, and anomalies that could be indicative of exocomets, aiding in making informed scientific conclusions.

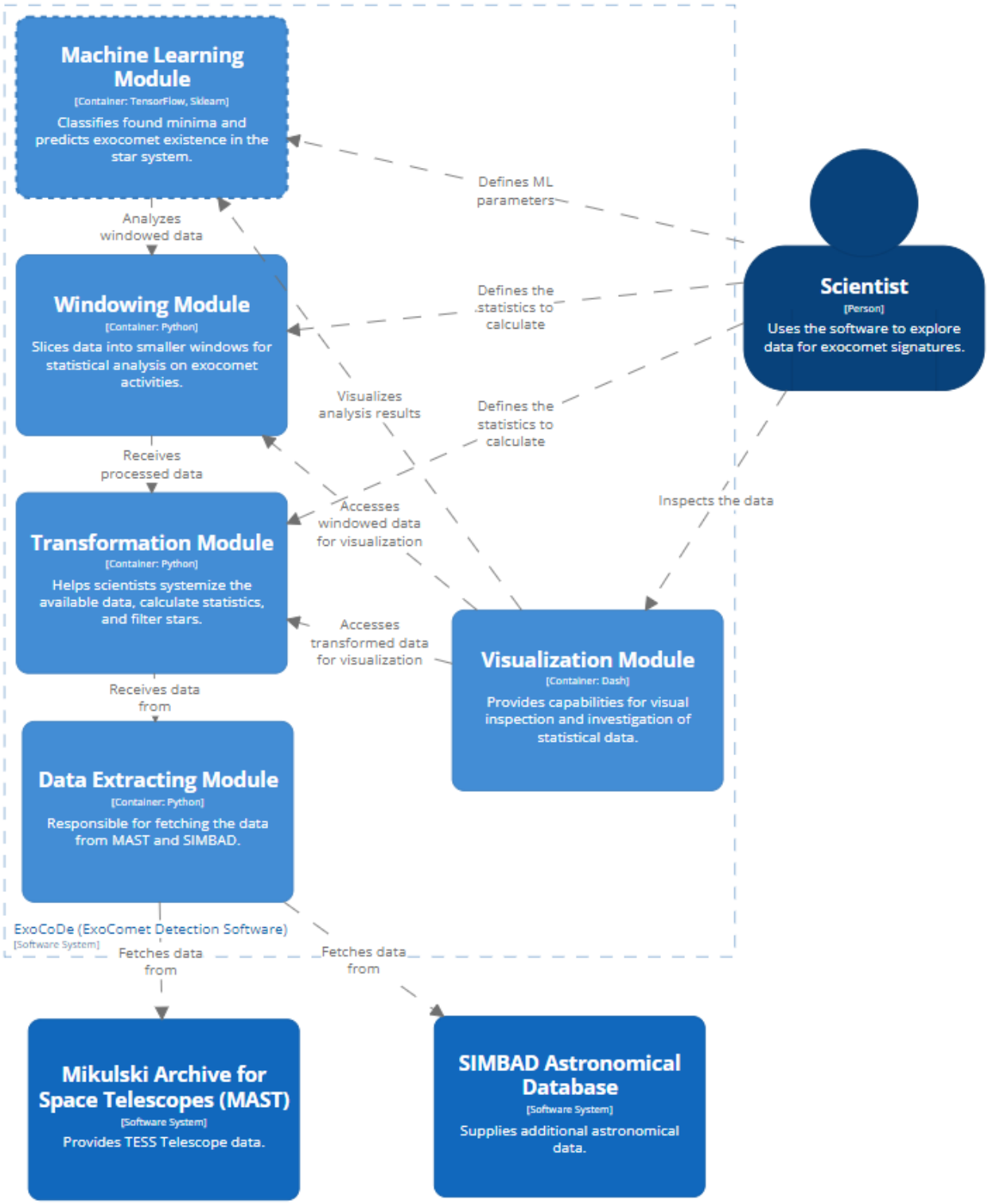


Fig 3. Container Diagram for ExoComet Detection software [15].

1.3.3. DB entities

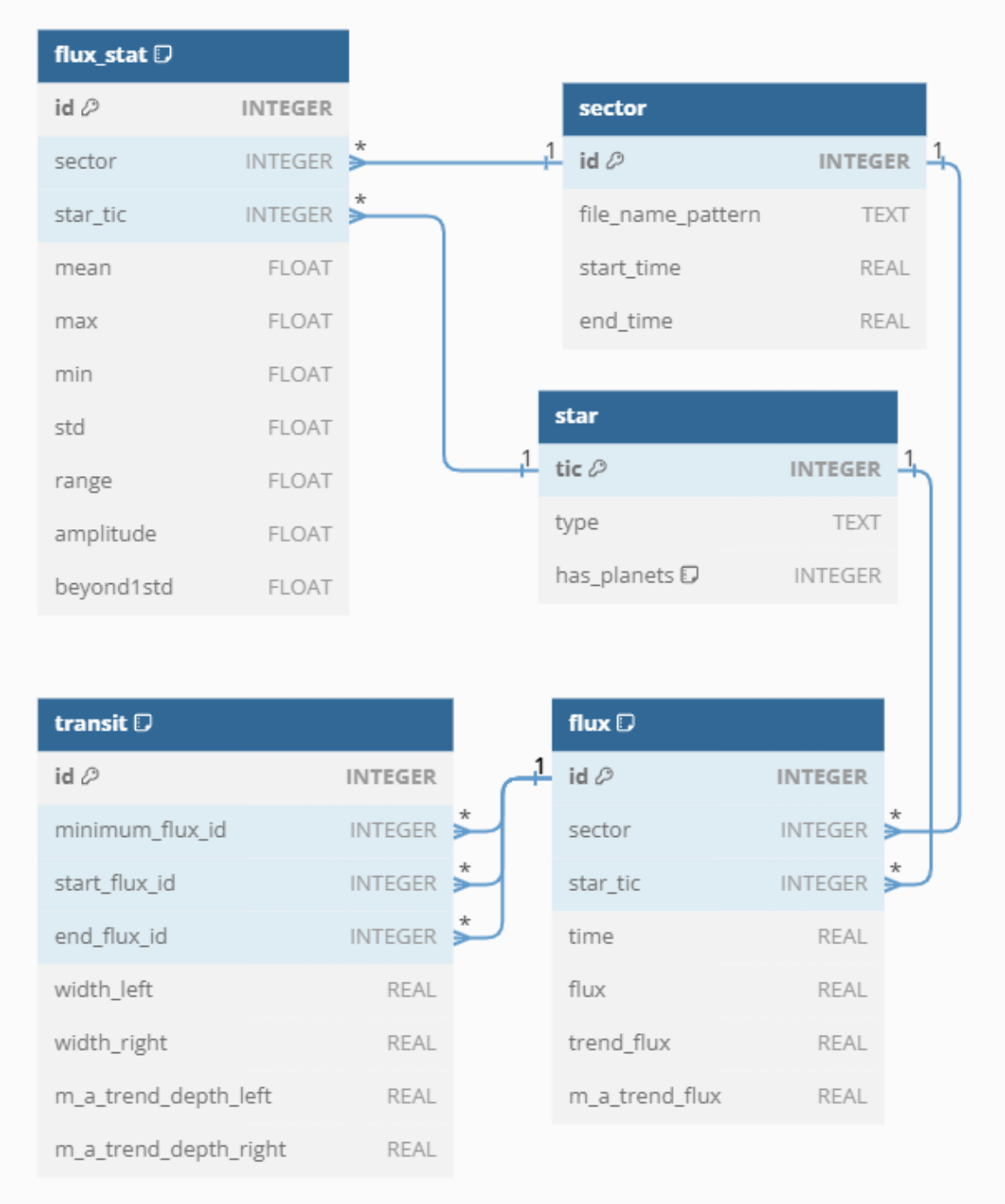


Fig 4. DB entities for the ExoComet Detection Software [19]**.**

SQLite was selected as the database for the project due to its simplicity and familiarity of the researchers.

Chapter 2. Data and Algorithms

2.1. The TESS Telescope Data

TESS Telescope collects a colossal amount of data from its cameras. It's translated from Full Frame images, by breaking down the data for individual stars, and results in the form of a time series. This data is generated by measuring the brightness of stars over regular intervals of time. For the project purposes, we will be using a 2-minute cadence. Each data point in the series represents the star's observed luminosity at a given moment, and the collection of these points over time forms a continuous record known as a light curve.

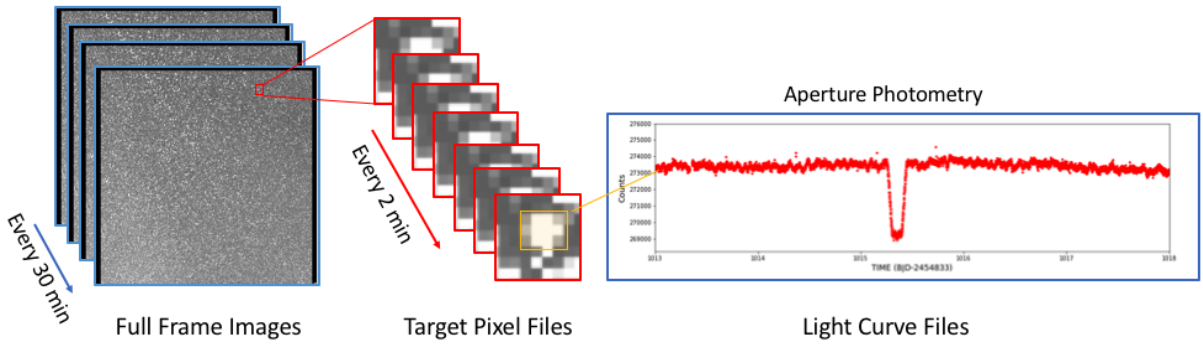


Fig 5. TESS Photometric data products [20].

The time series nature of TESS data is its most defining characteristic. It allows astronomers to observe changes in a star's brightness, which can indicate various astrophysical events. For example, a temporary dip in brightness might signal the transit of an exoplanet or an exocomet passing in front of the star from the sight of the observer from the Solar system. However, these light curves can be complex, with variations caused not only by transiting bodies but also by stellar activity: flares or spots, binary stars, pulsating, interference with other stars, etc. And this brings one of the challenges of the project. Moreover, exocomet transit events are even less possible to spot and more possible to confuse with other events or even signal noise.

The frequent sampling of the data allows for the detection of subtle and short-lived events, making it an invaluable tool for the discovery of exoplanets and exocomets especially.

TESS uses a unique observation strategy that involves dividing the sky into multiple sectors to systematically cover as much area as possible in its search for exoplanets and other celestial phenomena. Each sector is a large, rectangular segment of the sky, and TESS observes each sector for approximately 27 days. During this period, the satellite continuously monitors the brightness of stars within that sector. September 20, 2023, marked the close of Year 5 for TESS exploration, giving us data from 69 Sectors to work with for this project potentially in future, however, we will focus on only 1-2 sector during the Capstone project phase, as the time and resources are limited.

The richness of TESS's time series data provides the potential to reveal the dynamic processes occurring within distant star systems. However, the analysis of such data requires sophisticated techniques to distinguish meaningful patterns from noise and to identify the signatures of interest, such as those indicative of exocomets. Therefore, the time series nature of TESS data not only provides a window into the activities of distant stars but also presents an exciting challenge for data exploration and analysis.

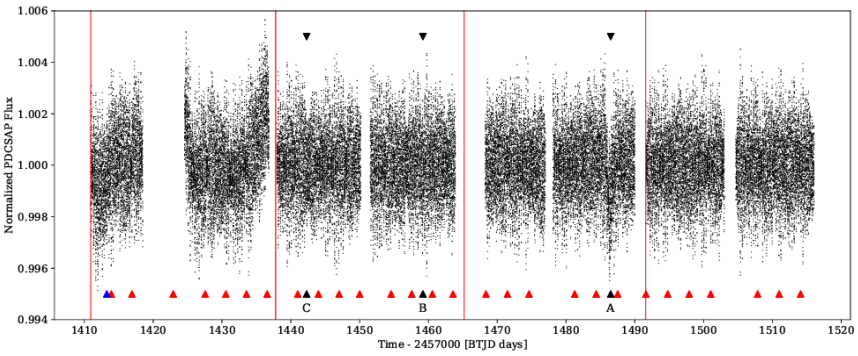


Fig 6. Full light curve of Beta Pictoris in Sectors 4-7 [7].

In order to have meaningful insights from the stellar data, some aggregational data preparations should be executed, as the flux signal is very noisy and there is no possibility to easily distinguish transit events from the raw data. Thus, this will be one of the key tasks of the data preparation step of the project.

The main feature of any transit event in front of the star is the dip in the light curve, which represents the reduction in the star's brightness as another body blocks a portion of its light. The depth of the dip is proportional to the size of the body relative to the star; larger bodies block more light and cause deeper dips.

The profile of planetary transit is characterized by a symmetrical, U-shaped dip in the light curve. The symmetry reflects the uniform motion of the planet as it crosses the star's disk. During the full transit, when the planet is entirely in front of the star, the light curve typically shows a flat bottom.

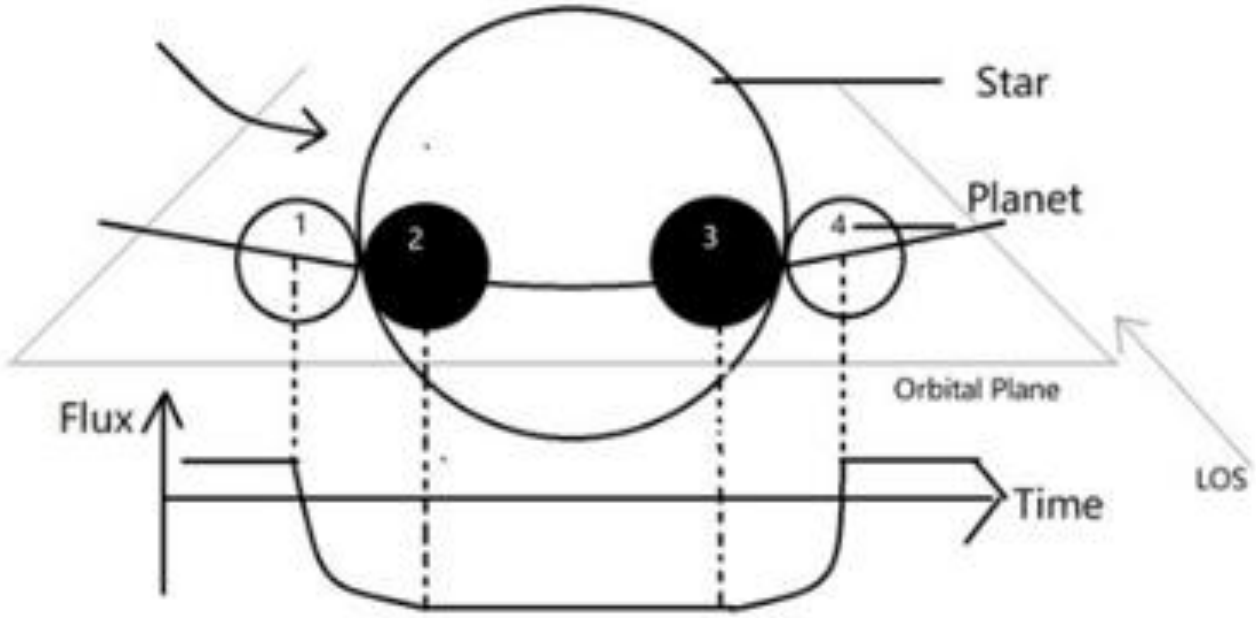


Fig 7. A disc-crossing transit of a planet. The stellar flux drops from 1 to 2, remains a constant from 2 to 3, and returns to normal flux level from 3 to 4 [21].

The profile of a cometary transit, particularly for exocomets, presents a different and more complex signature on a light curve compared to a planetary transit.

Let's define key features of a cometary transit profile. In accordance with Kennedy, et al. [8], unlike the symmetrical planetary transits, cometary transits often display an asymmetrical pattern. This is because a comet's coma (the cloud of gas and dust surrounding its nucleus) can create a shadow on the star. This is why cometary transits can also be longer in duration compared to planetary transits.

While planetary transits are typically regular and periodic, comets' transits can be irregular and less predictable, it also can change its appearance due to outgassing, or changes in the coma and tail.

The overall decrease in brightness during a cometary transit is usually smaller than that of a planetary transit. This is because a comet, even with its extended coma and tail, generally covers a smaller area of the star's disk compared to a planet.

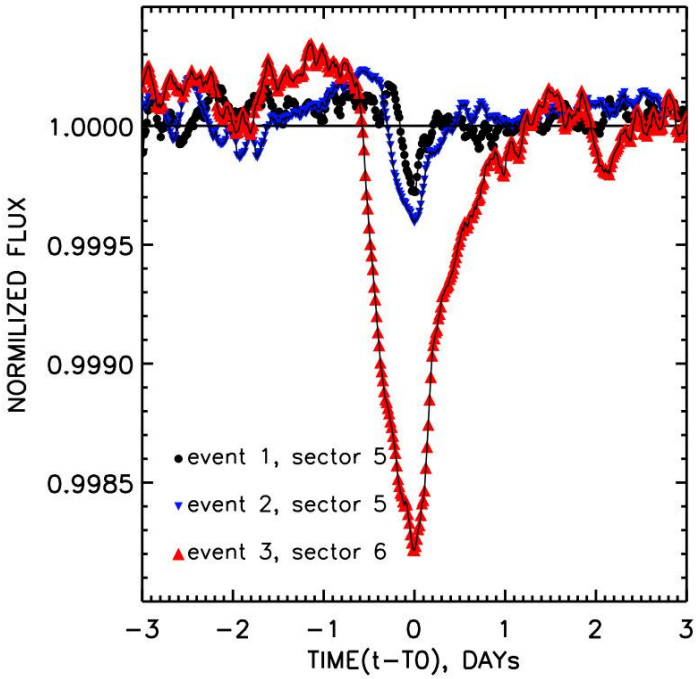


Fig 8. Asymmetric dips previously found in the TESS light curves of Beta Pictoris star in sectors 5 and 6 (Zieba et al. 2019) [1, 7].

In order to solve the task of the project, firstly, we need to do an exploratory data analysis to define which stars data could be taken into account further to find the exocometary activities. This will be done by applying both statistical and astrophysical algorithms.

The changing nature of star data provokes us to reduce data complexity by removing some of the flux characteristics, like star fluctuations, binary stars data, etc. This could be done by applying Fourier or Lomb–Scargle methods, based on the research of Kennedy [8].

However, for the initial part of the project it was decided to focus on the stars without high fluctuations by collecting the information about type of the stars from a SIMBAD Astronomical Database [17], and applying the method described below on this limited amount of stars.

Furthermore, some of the sectors have corrupted segments due to some telescope calibration events, which could be filtered out in the further stages of the project. Without this filtration, the system could give more false positive results of found exocomets, and we need to take this into account while interpreting the results of the research.

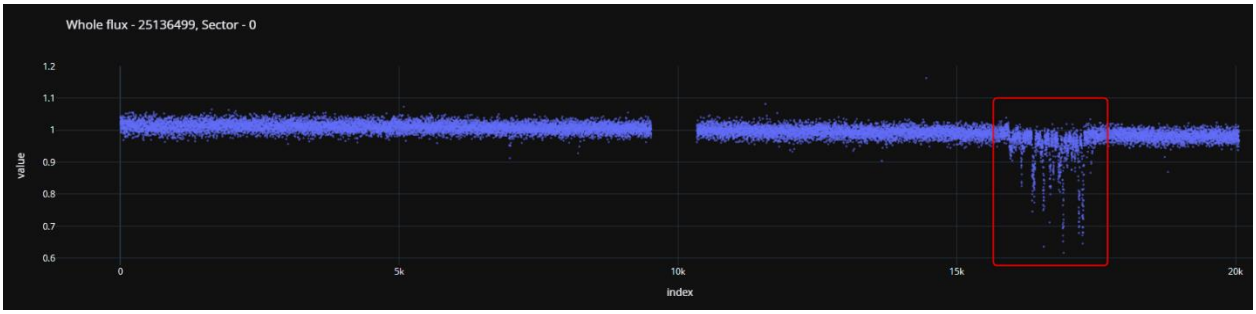


Fig 9. Corrupted data requiring deletion.

2.2. Data manipulation pipeline algorithms

To address the project's requirements, we must perform certain operations on the data. Different forms of data are processed by different modules, specifically:

Module	**Data and task solved**
Data Extracting Module	Raw flux data Normalized flux data
Transformation Module	Flatten flux data Moving average flux data
Windowing Module	Moving average flux difference Finding potential transits minimums and surrounding windows
Machine Learning Module	Characterizing the parameters of those potential transits

Table 1. Pre-processing modules and their tasks

Let's explore different forms of data and the algorithms of its retrieval.

2.2.1. Raw and normalized data

The time-series for the flux are saved in specific files with the extension **'fits'**, and they could be retrieved from the Mikulski Archive (which was already done by the astrophysicists on the project), and those files are our initial point to start with. The data for the different stars has different order of magnitude for the flux values, depending on the brightness of the star, and sometimes it brings more complexity to the analysis of the star's features.

To compare the flux of the stars between themselves we should normalize the flux, and we could exploit LightKurve library base method LightCurve.normalize() to do it [22], which is designed specifically for astrophysical data and refers to scaling the light curve data so that its median is 1. This type of monotonous normalization is often used in astrophysics to compare the brightness of celestial objects over time, where the actual brightness is less important than the relative change in brightness.

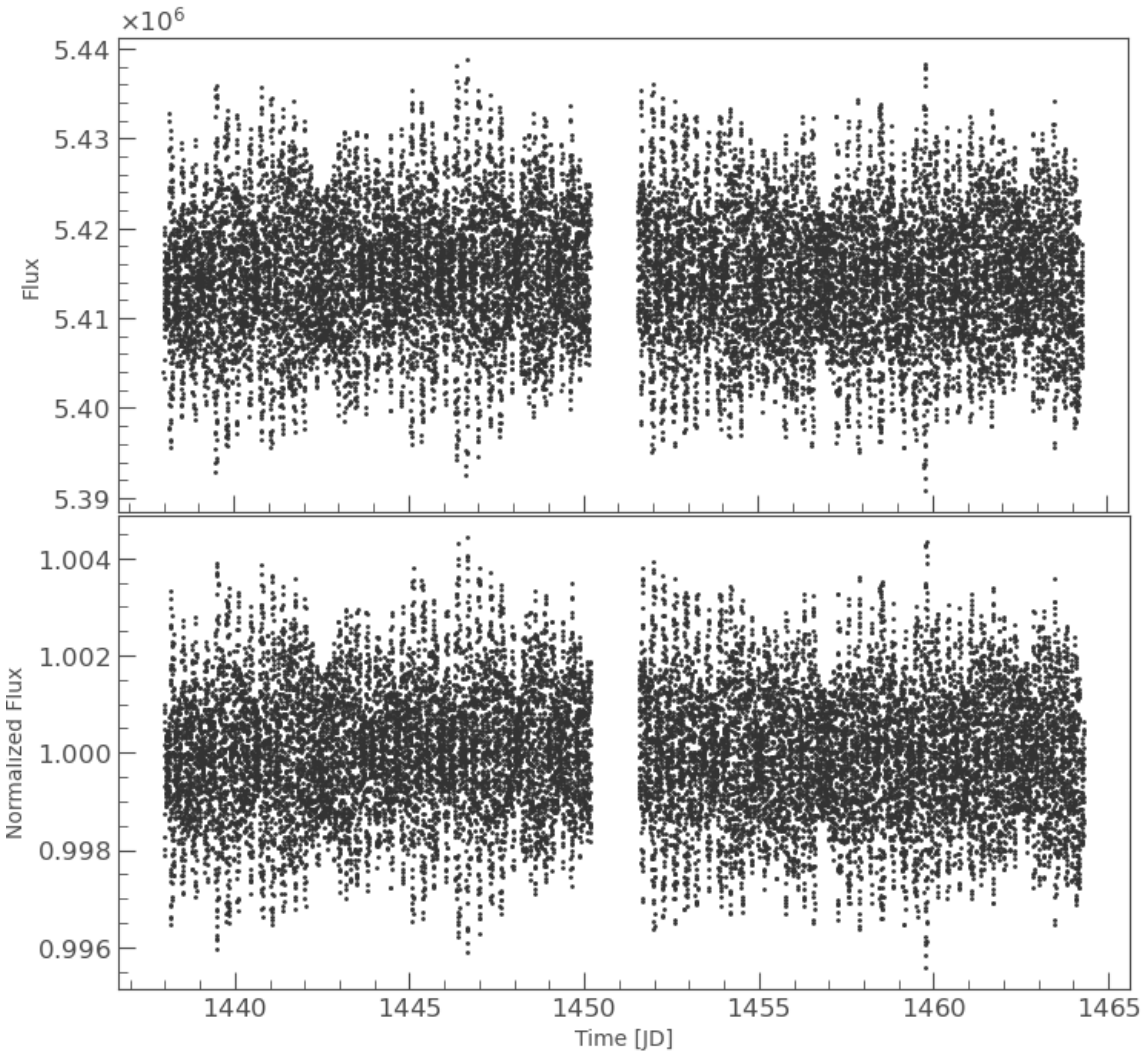


Fig 10. Scatter plot of raw data for the flux of one star, Beta-Pictoris star from sector 5 used as example here and after (upper plot), and its normalized representation scaled around 1 (lower plot).

2.2.2. Flatten/trend data

As we can see from the plot above, there is not so much useful information which could be taken from the plots now, as it's very noisy. So, we need to get rid of this noise and provide a flatten interpretation of the signal from the star. There is another astrophysical library called **wotan** and its method, called **flatten,** which could be exploited for this task. Basically, the method "removes low frequency trends in time-series data" [10]. The library's name is inspired by Odin, a deity from Germanic mythology. The Old High German version of Odin is "Wōtan," famously depicted wearing a horned hat, which resembles the silhouette created during a transit event of an exoplanet.

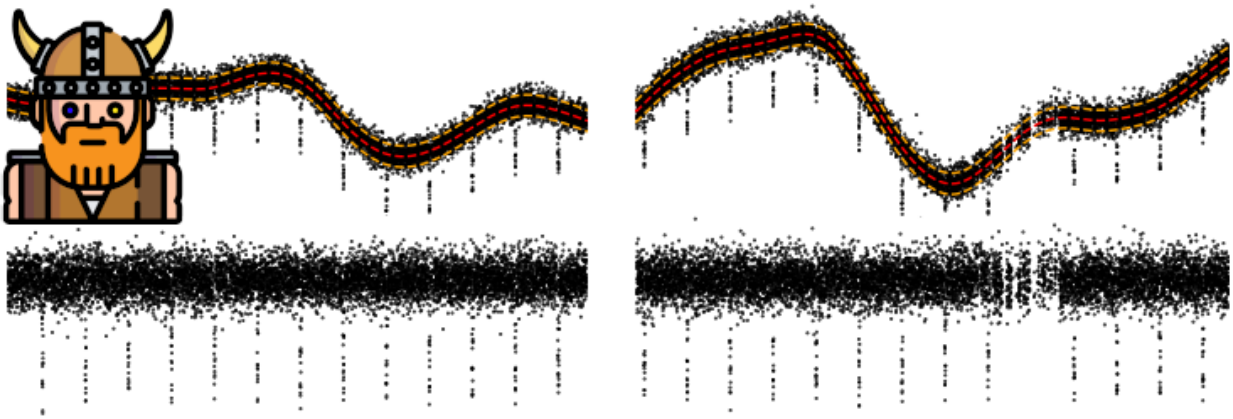


Fig 11. Wōtan offers free and open-source algorithms to automagically remove trends from time-series data [23].

There are several parameters used for the de-trending, one of which is **window_length** for the flattening. By the empirical means it was decided that value **0.3 days** serves the task best, however, it could be parametrized and investigated further. Also, there are different methods for flattening, and we are using the default **biweight** method here. After applying this flattening algorithm, we will start seeing profiles of the star signal, and can start spotting the transit events in the data.

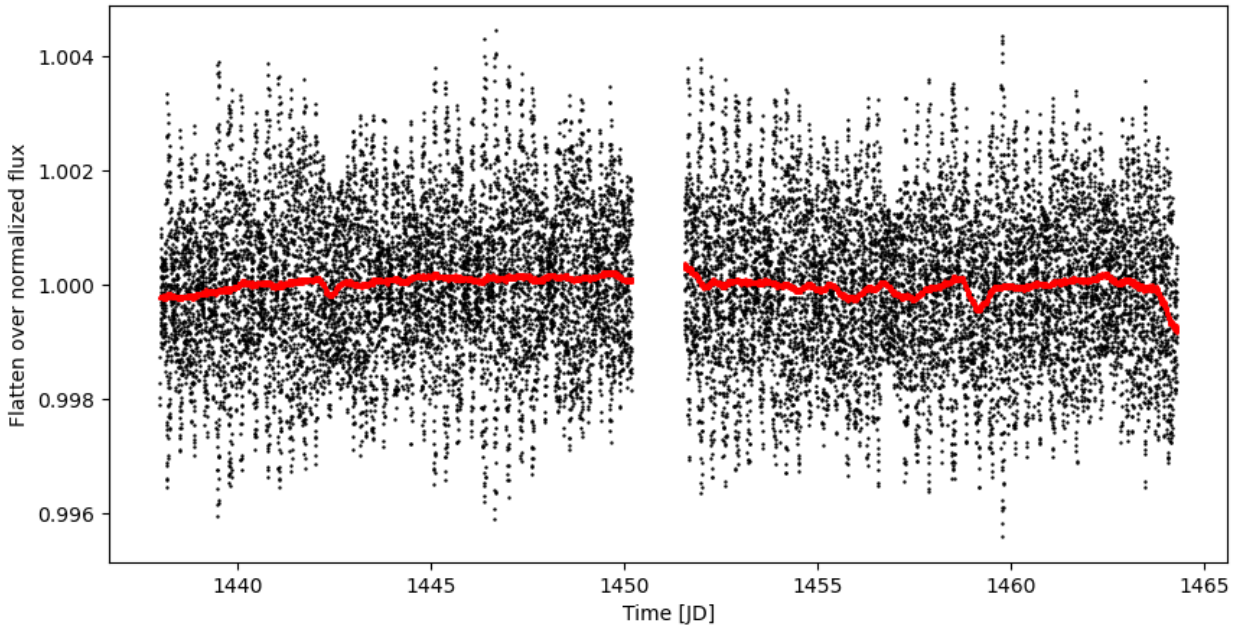


Fig 12. Scatter plots of normalized data around 1 for the flux of one star and its flattened representation (in red).

2.2.3. Moving average trend data

Our data is de-trended now, however, there are still some fluctuations, which could interfere with our goal of finding exocomet transit. To avoid this, we could apply binning mechanism to smooth the data with a moving average algorithm. For our purposes we've decided to use **the centered method** for the rolling windowing, calculating the mean for some window to the left and to the right, so our binned data will not be shifted to some side. To explore the process of binning we need to zoom in a bit for some segment to see the difference between de-trended and moving average data. In our case, as the Beta-Pictoris star in 5^{th} sector has the events which are considered as exocomet transits [7], we will look at those specific events.

Empirically, the window size was decided to take as **90 points**, however this value could be further optimized to get better results.

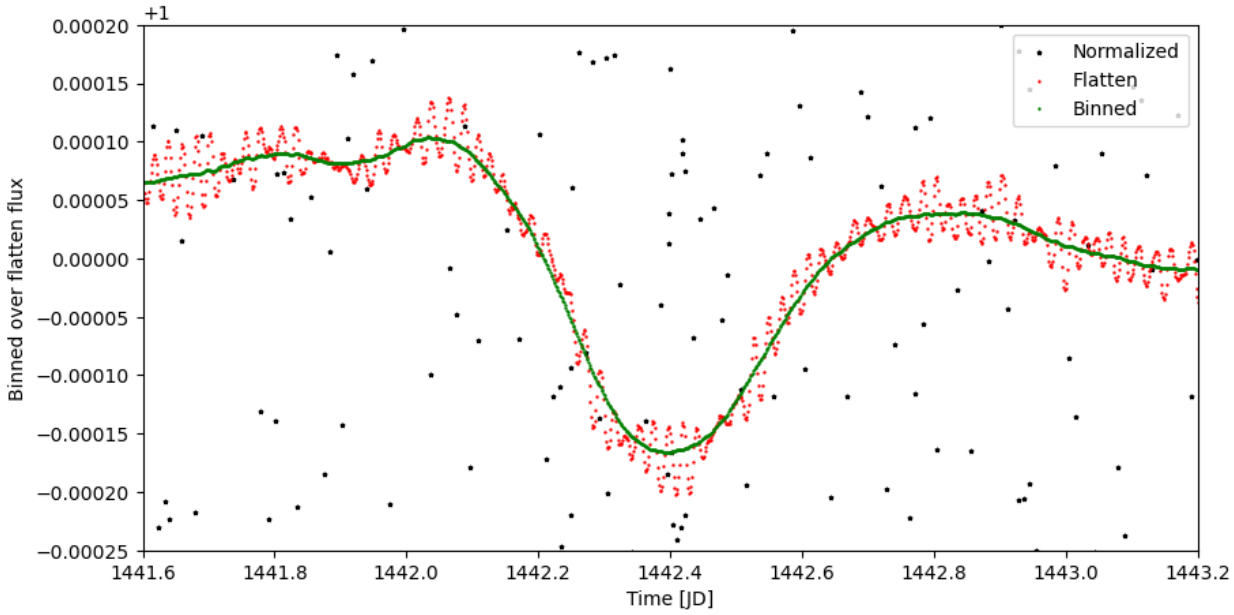


Fig 13. Scatter plots of normalized flux data (in black) of one transit event, its flatten trend representation (in red) and moving average representation (in green).

As you can see, we reduced the fluctuations by saving the overall profile of the transit event, which could help us find such events for other stars.

The centered moving average at time ***t*** is given by:

$$\bar{f}(t) = \frac{1}{w} \sum_{k=-\frac{w}{2}}^{\frac{w}{2}} f(t+k),$$

where:

- $\bar{\boldsymbol{f}}(\boldsymbol{t})$ is the centered moving average trend flux at time ***t***,
- $\boldsymbol{f}(\boldsymbol{t})$ is the trend flux value at time ***t***,
- $\boldsymbol{w}$ is the window size, representing the number of points included in the moving average calculation,
- $\boldsymbol{k}$ is an index variable that runs from $-\frac{\boldsymbol{w}}{\boldsymbol{2}}$ to $\frac{\boldsymbol{w}}{\boldsymbol{2}}$, indicating the range of data points included in the average, centered around the current point ***t***.

2.2.4. Moving difference data

To find some time-series intervals, which could be considered as transit events we can calculate the difference between neighboring points of binned data and look for the segment where considerable amount of data with negative trend is followed by positive trend. For our example of Beta-Pictoris exocomet transit event we can see it on the plot below:

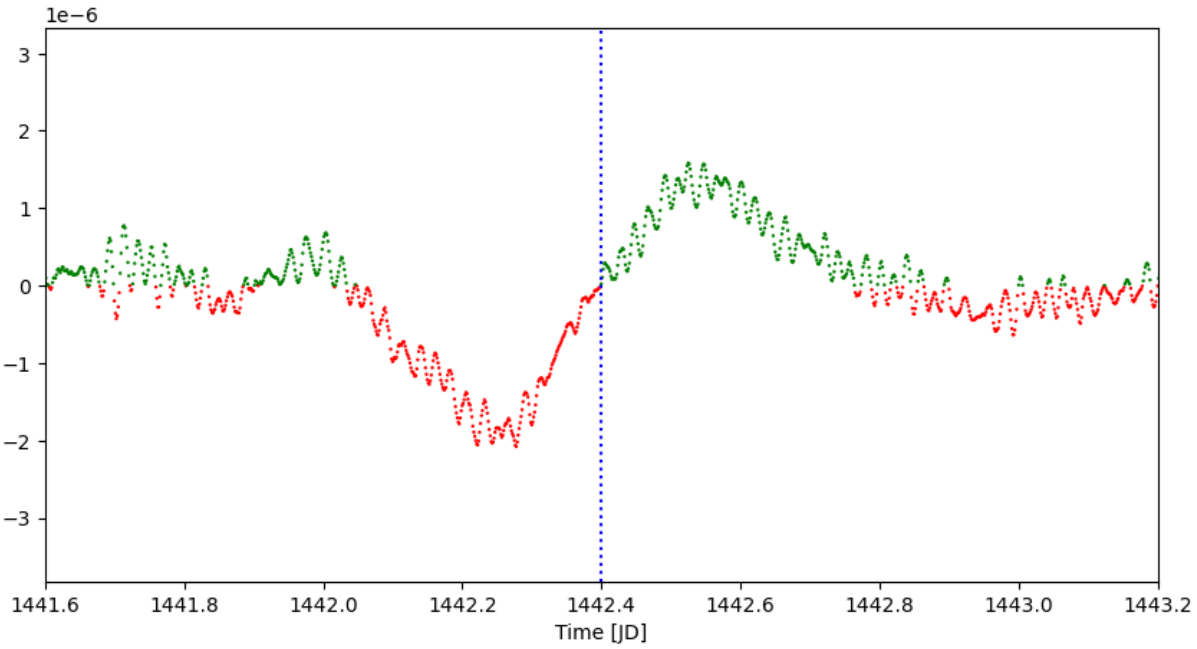


Fig 14. The differenced data of moving averages for trend flux with a lag of 1 point. Negative trend - in red and positive - in green. The point of minimum is reflected with dotted blue line.

The differenced data of moving averages for trend flux is given by:

$$y(t) = \bar{f}(t) - \bar{f}(t-1),$$

where:

- $\boldsymbol{y(t)}$ is differenced data,
- $\boldsymbol{\bar{f}(t)}$ is the centered moving average trend flux at time ***t***,
- $\boldsymbol{\bar{f}(t-1)}$ is the centered moving average trend flux for lagged time ***t-1***.

This way we have reflected our signal data in the form of moving trends, which could be further analyzed for the presence of windows, that could be considered as transiting events, specifically with exocomets. The difficulty for the task is that the windows of the transits could be of different width, so we need to look for the different combinations of width for left and right wings of the transit event. However, we can focus on the points where the sign of the trend has changed from negative to positive, thus the number of calculations will decrease significantly.

2.2.5. Looking for appropriate windows

Among those points where our trend is changing its sign, we need to calculate the maximum width of the negative trend of differenced moving average trend flux to the left, allowing **5%** of the fluctuations, meaning we will be looking for **95%** negative trend. We are interested in the events width of which is between **2** and **0.3** days (approximate duration of the half of the transit). After that, we limit our calculations to only those points, where such appropriate negative trends are found from the left side and calculate positive trends to the right from such points of flux minimum.

$$w_{\text{left max}}(t) = \max\left\{w \in W_{left} \middle| \frac{1}{w}\sum_{i=t-w+1}^{t} 1_{\{y(i)>0\}} \leq 0.05\right\},$$

where:

- $\boldsymbol{w_{\text{left max}}(t)}$ is the maximum width of the window to the left of point t, where the conditions are satisfied,
- $\boldsymbol{W_{left}}$ is the set of possible window widths {2, 1.95, 1.90, …, 0.30}, decreasing by steps of 0.05,
- ***y(i)*** represents the differenced data at point ***i***,
- $\boldsymbol{1_{\{y(i)>0\}}}$, is an indicator function that is 1 if ***y(i)*** is positive and 0 otherwise,
- The sum $\boldsymbol{\sum_{i=t-w+1}^{t} 1_{\{y(i)>0\}}}$ calculates the count of positive values within the window,

- $\frac{1}{w}$ normalizes the sum to the window width to find the proportion of positive values,
- and the inequality ≤ 0.05 ensures that the window contains no more than 5% positive values, allowing for at least 95% negative values within the window.

The left and right sides of the transits could be of different width, with the right ascending one potentially wider, as comet dust takes some time to dissolve and reveal the full star flux. Thus, we are interested only in points where the width of the right side of the potential transit width takes 100% to 400% of the left side, providing us with even less points to investigate manually further.

$$w_{\text{right max}}(t) = \max\left\{w \in W_{right} \middle| \frac{1}{w}\sum_{i=t}^{t+w-1} 1_{\{y(i)<0\}} \leq 0.05\right\},$$

where:

- $\boldsymbol{w_{\text{right max}}(t)}$ is the maximum width of the window to the right of point t, where the conditions are satisfied,
- $\boldsymbol{W_{right}}$ is the set of possible window widths {4 * $\boldsymbol{W_{left}}$, …, $\boldsymbol{W_{left}}$}, decreasing by steps of 0.05,
- ***y(i)*** represents the differenced data at point ***i***,
- $\mathbf{1}_{\{y(i)<0\}}$, is an indicator function that is 1 if ***y(i)*** is negative and 0 otherwise,
- The sum $\sum_{i=t}^{t+w-1} \mathbf{1}_{\{y(i)<0\}}$ calculates the count of negative values within the window,
- $\frac{1}{w}$ normalizes the sum to the window width to find the proportion of negative values,
- and the inequality ≤ 0.05 ensures that the window contains no more than 5% negative values, allowing for at least 95% positive values within the window.

Chapter 3. Exocomet Event Analysis and Visualization

3.1. Finding ingress and egress points of transit event with machine learning

To identify the start (ingress) and end (egress) of potential transit events, where a comet crosses in front of a star and then exits, we examine the changes in light intensity (differenced data we have). We look for where the trend in light difference becomes negative on the left for ingress and positive on the right for egress. However, this method can lead to misinterpretations due to minor fluctuations on either side of the transit, causing frequent sign changes.

A more reliable approach involves using a shifting linear approximation of the time-series data. By applying a Linear Regression algorithm and detecting when the slope nears zero on either side, we accurately pinpoint ingress and egress points.

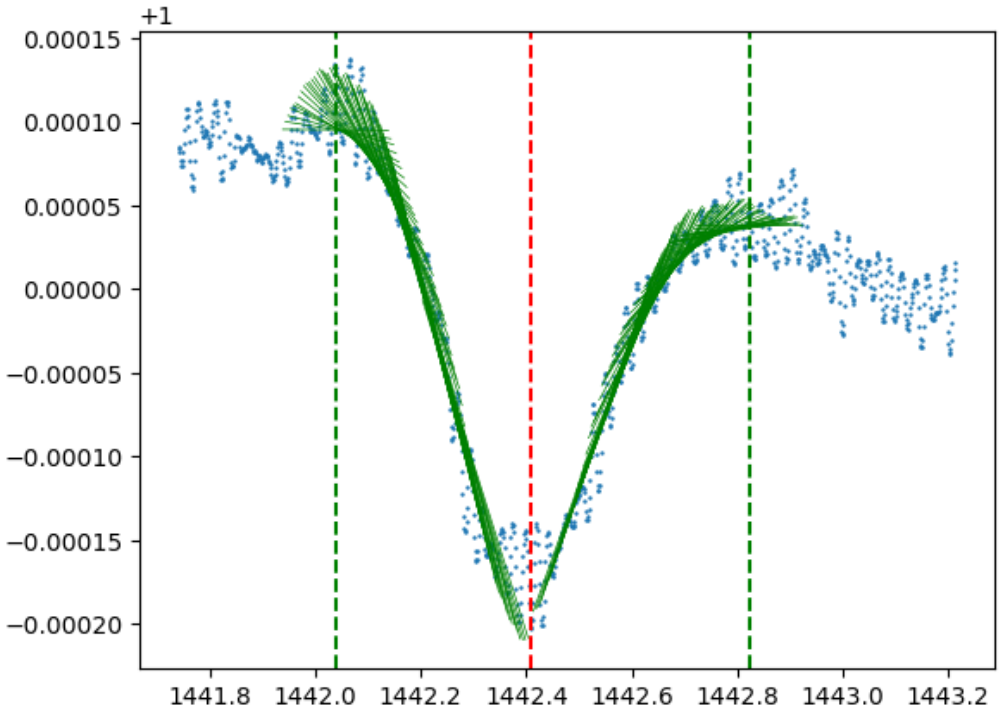


Fig 15. Linear approximations to find the ingress and egress points (vertical dashed green lines) moving away from point of minimum (vertical dashed red line).

For this we are using LinearRegression algorithm from sklearn library. With the step size ***Δt,*** we are moving from the point of minimum and calculating the slopes of the approximated linear regression for the data withing a particular centered window.

$$m = LinearRegression\left(flux\left[t - \frac{W}{2}, t + \frac{W}{2}\right]\right).slope,$$

if we are moving left from current ***t***, and $|m_{left}| < \theta$, then $t_{ingress} = t$,

if we are moving right from current ***t***, and $|m_{right}| < \theta$, then $t_{egress} = t$.

Where:

- ***W*** is the time window size.
- ***Δt*** is the step size used for moving through the time series.
- ***θ*** is the slope threshold for detecting ingress or egress.
- ***t*** is the current time point being evaluated.
- ***m*** is the slope of the linear regression line fitted to the data within the window centered at ***t***.
- $\boldsymbol{m_{left}}$ and $\boldsymbol{m_{right}}$ represent the slope of the linear regression to the left and right of the time point ***t***, respectively.
- $\boldsymbol{t_{ingress}}$ and $\boldsymbol{t_{egress}}$ are the required points of ingress and egress, respectively.

The values for the constants were decided to take as the following:

$$W = 0.3, \quad \Delta t = 0.005, \quad \theta = 0.00001$$

3.2. Web interface to explore the data

So, the main interface to explore the results of the transits detection will be the web application with several parts. It's implemented using the Dash library, described before, which relies heavily on Flask as its underlying framework. Flask provides the foundational web server capabilities, and Dash by Plotly is focusing more on the interactive dashboards.

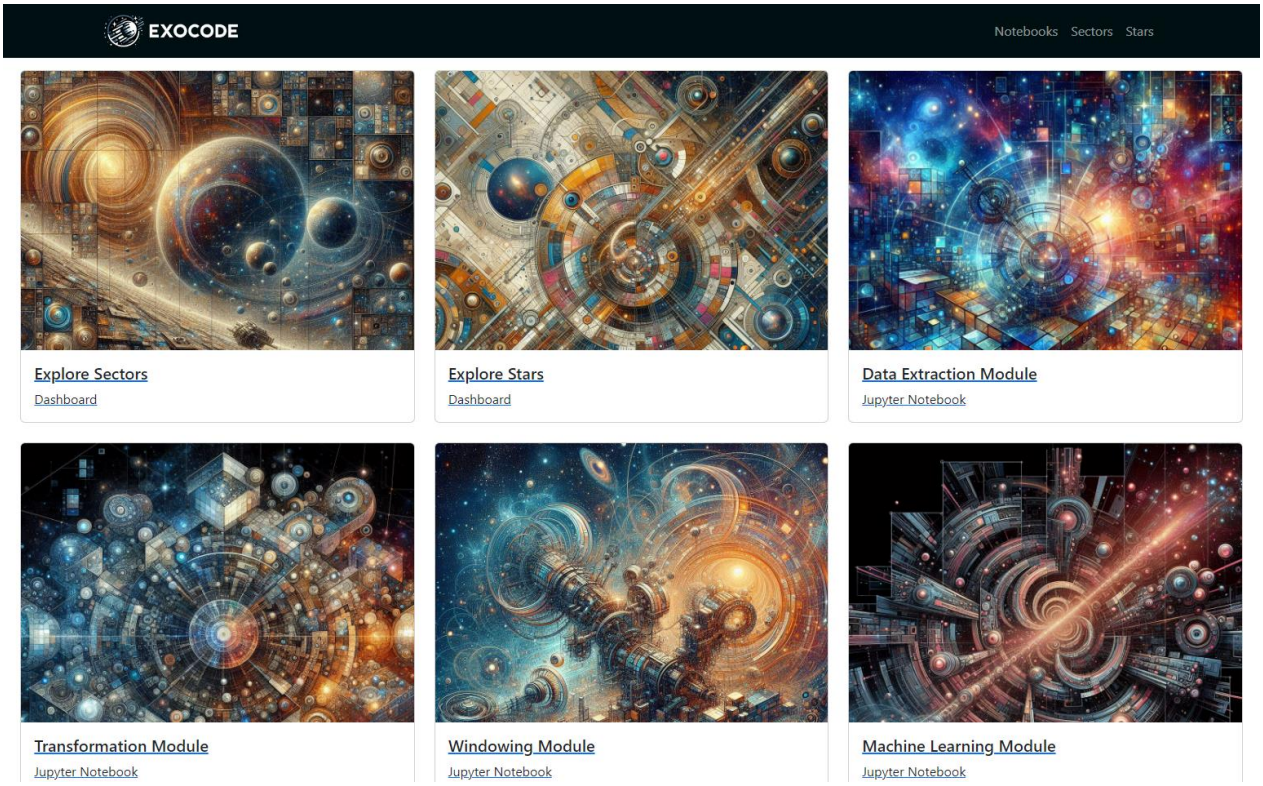


Fig 16. Main interface of ExoCoDe system.

Four main modules of the system, described above, namely: Data Extraction, Transformation, Windowing, and Machine Learning, will be represented as separate Jupyter Notebooks, running on the separate Jupyter server, which would be accessible for the scientists to apply the necessary changes to the logic, expand it with additional parameters, track the execution of the necessary data pipeline steps etc.

This way, the different modules could work in parallel, using different instances of Python kernel, so using the advantages of multiple processors.

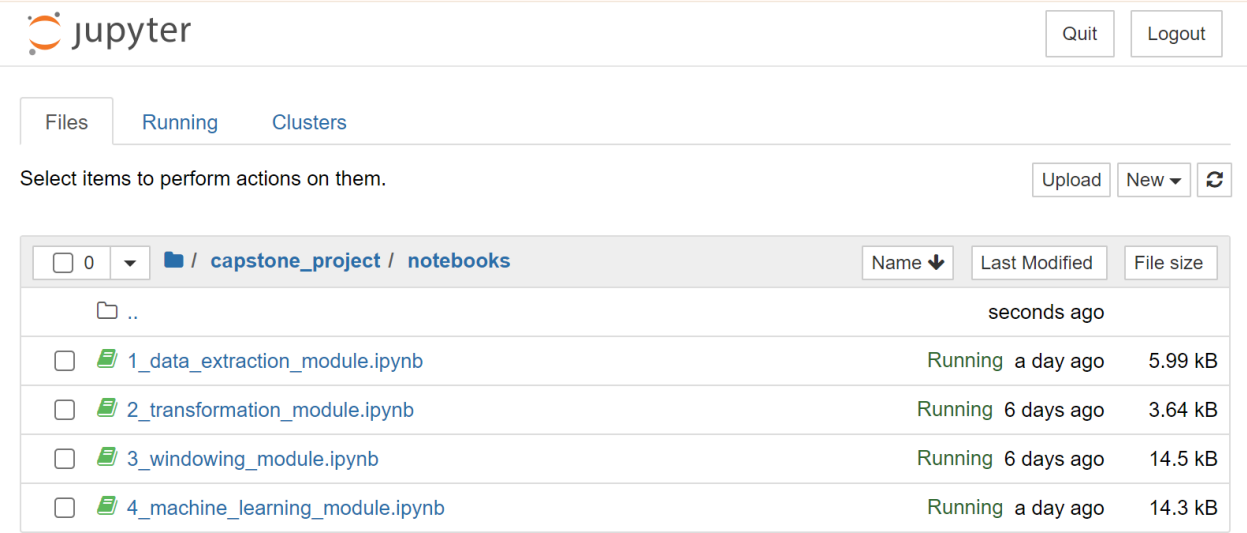


Fig 17. Jupyter Notebook interface for interaction with modules, manipulating with TESS telescope data.

The implementation of each module here is the Python code representation of the algorithms, which we've already described above. Notebooks have the possibility to change the different parameters to some extent, to finetune the execution of the algorithms. However, some common code is intentionally extracted to separate files.

The visualization part is split into two pages: Sectors and Stars. Let's investigate them more closely.

3.2.1. Sectors

The Sectors page provides a view of our data about the stars in the form of an interactive scatter plot, which shows some defined statistics for all processed stars in a sector. There is a possibility to select the type of statistical value for each axis. The star types here are represented by different colors, and the user can filter only those which are of interest, by clicking on the type on the right panel.

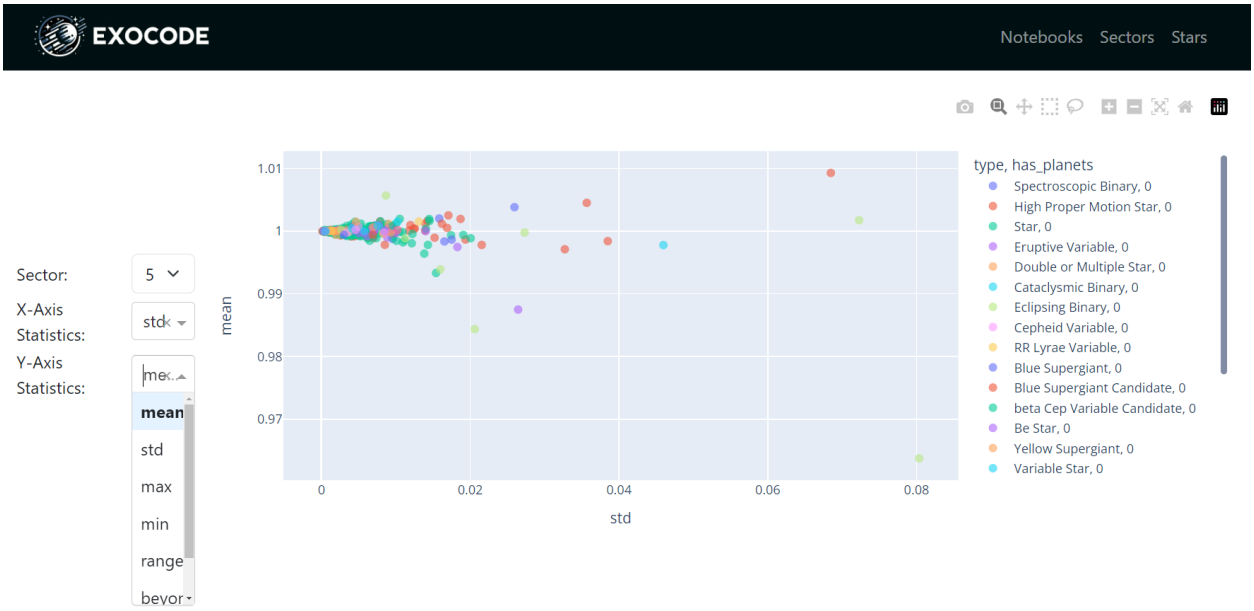


Fig 18. Scatter plot for the stars in the TESS Sector, X and Y axes are representing different chosen statistics.

Such a view helps scientists to find some anomalies in the set of stars and filter them out for further analysis. Different statistics help to look at luminescence data of the sample of the stars from different perspectives.

Along with standard statistics of minimum, maximum, mean, and standard deviation values, we've selected to show range – calculated as difference between min and max values; amplitude defined as the half of the difference between the median of the maximum 5% and the median of the minimum 5% magnitudes; beyond1std

quantifies the proportion of data points that fall outside the range of one standard deviation from the weighted average, serving as a measure of the data's variability.

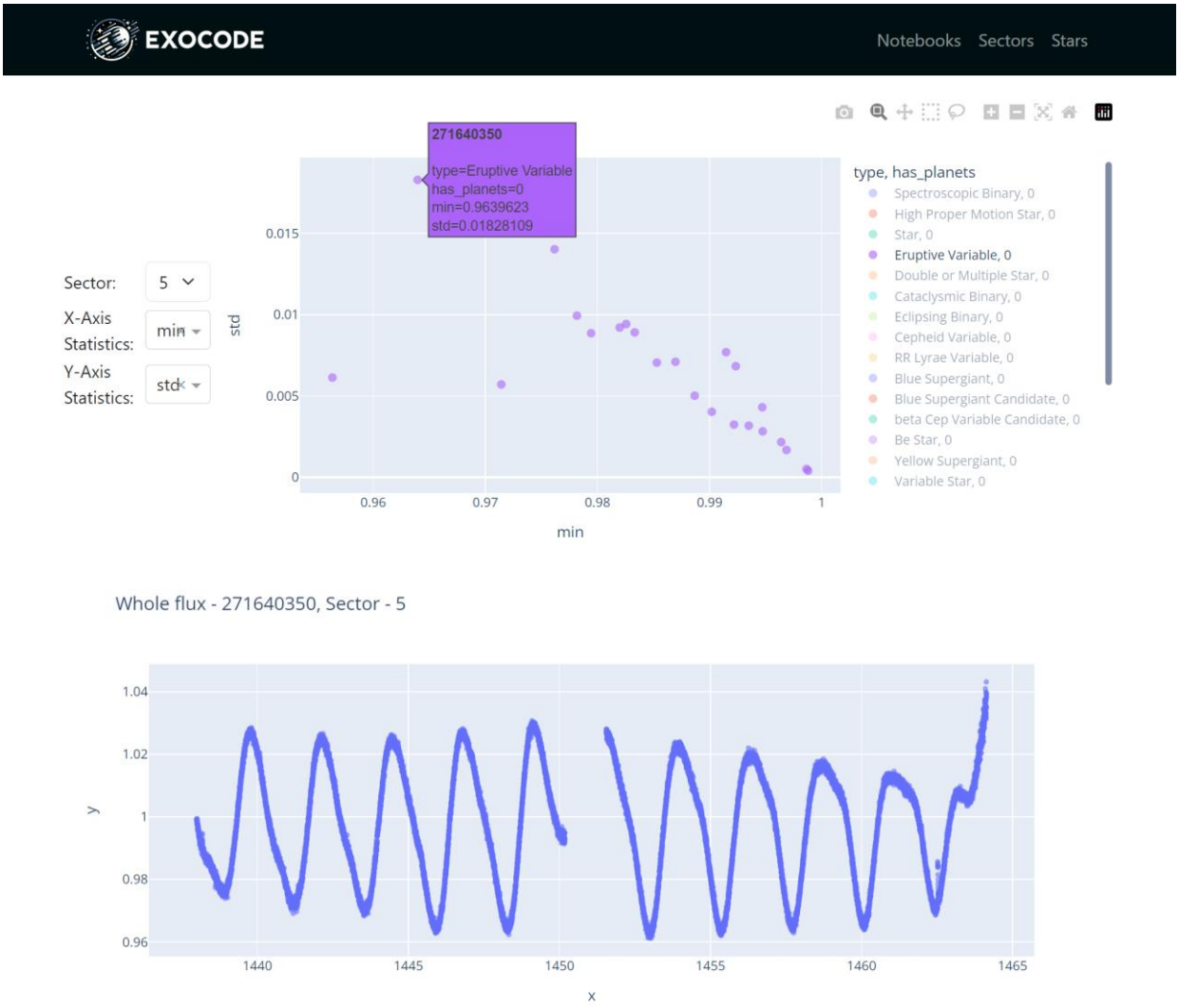


Fig 19. Filtered only Eruptive Variable stars on the graph above. Selected point for a star's statistics displays the plot for its flux below.

The Sectors page could provide the overview of the particular timeframe explored by the telescope, and could help finding the anomalies, which are relevant to all stars observed in this particular sector, so filtering those anomalies out from the list of potential transit events.

3.2.2. Stars

The Stars page is used for the visualization of the information about stars in a particular sector. For this purpose, we are using several Dash scatter plots on the same graph, specifically to represent:

a) the raw normalized data,

b) the trend data received with using Wotan library,

c) the smoothed flux, received with a moving average algorithm.

The plot is also interactive and could be zoomed in to see the parts of the flux and used to evaluate the fluctuations in the star signal.

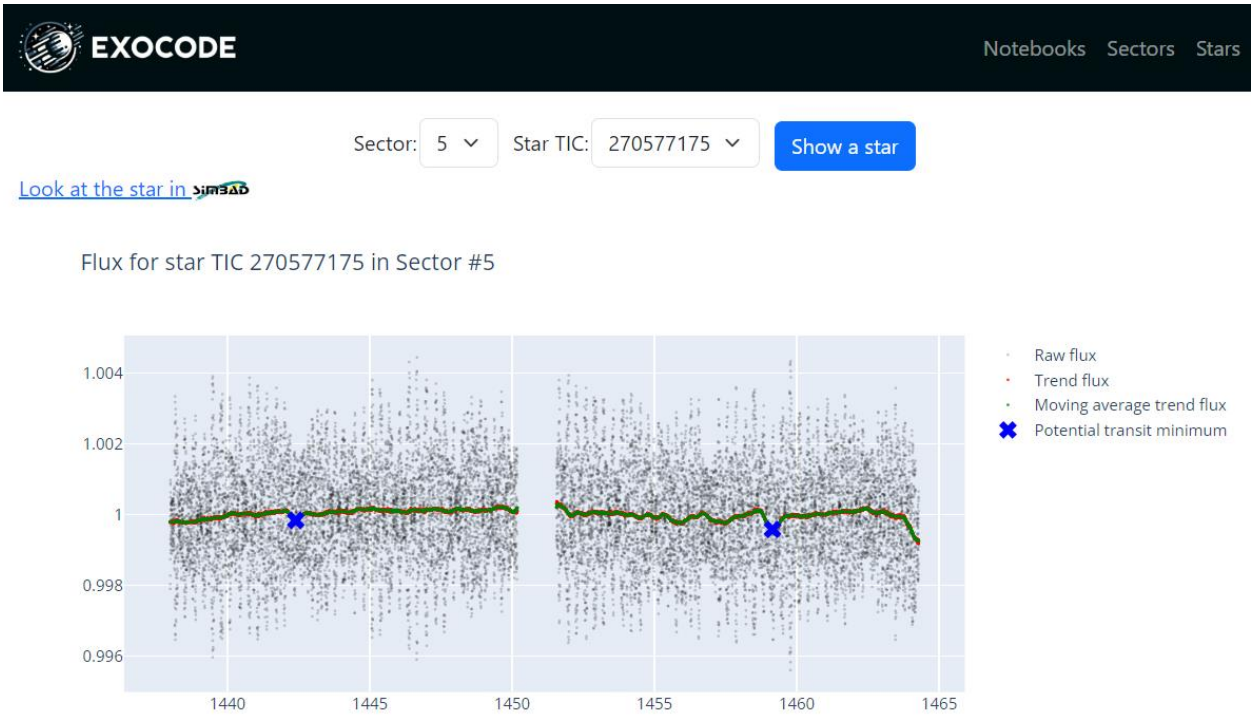
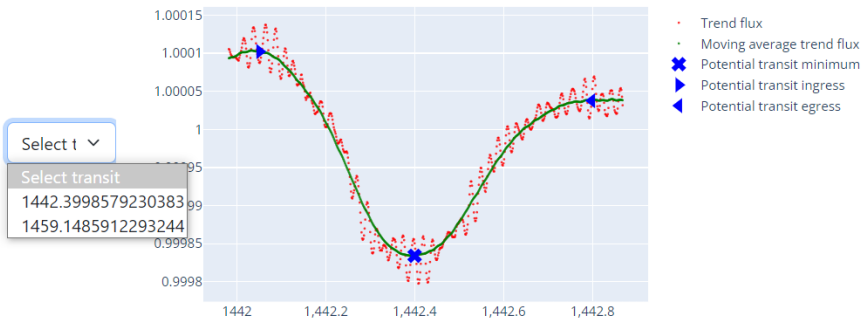


Fig 20. Exploration of the different representations of the flux signal for a particular star in a sector, with blue markers showing potential transit events.

The user has the possibility to select the TESS Sector, where the star of interest was observed. After that, the star id (TIC) should be selected. Clicking on the "Show a star" button will lead to the reload of the scatter plots.

If any transit events were found for this star with the Jupyter Notebook modules, they will appear on the star graph as blue cross markers, and the transit events will appear below the star graph in zoomed-in representation, like on the figure below. The user can also switch between all the transits found to investigate it in more detail.

The parameters of the transit are also represented in the table near the transit plot, showing the values for ingress, minimum, and egress points; width of the whole transit, entering, and exiting phases of the transit; as well as depth for both sides of the transit.



Transit parameter	Value
Ingress point (t)	1442.0512424692952
Minimum point (t)	1442.3998579230383
Egress point (t)	1442.7984738722457
Transit width (t)	0.7472314029505469
Entering width (t)	0.3486154537431503
Exiting width (t)	0.3986159492073966
Depth from ingress	0.00026817903828924017
Depth from egress	0.0002035951475733988

Fig 21. Exploration of the transit events, highlighting its main points: Ingress, Minimum, Egress, and showing basic calculated parameters

Conclusions and Future Steps

The result of the project is a system that offers automated capabilities for identifying potential transit events of exocometary activities. This system has successfully pinpointed exocometary transit events that were previously identified in research on the topic, demonstrating its effectiveness for similar explorations. By applying statistical and machine learning algorithms, the ExoCoDe application significantly reduces the time required for exploring exocomet transits. This could lead to breakthroughs in understanding exocomet activity and potentially answer the longstanding question of whether comets brought life to Earth long ago.

The novelty of the proposed solution lies in shifting the focus from analyzing a limited number of stars to broadly scanning the available data in order to identify exocomet candidates through a data manipulation pipeline. Furthermore, the proposed web interface offers an option to explore data related to TESS telescope sectors. It highlights various types of stars with their statistical parameters, as well as provides detailed information on specific stars, including any potential transit events discovered.

In this manner, the system fulfills its functional requirements within the constraints of the project, offering not only the capability to detect exocometary activities but also a user-friendly interface for exploring the results.

Furthermore, there are several areas that could be enhanced: refining the filtering process for observations based on the noisiness of the signal, eliminating artifacts and corrupted data from sectors to prevent false positive detections, and identifying more transit characteristics, which could lead to a more focused exploration of exocomets. Additionally, the issue of duplicate transit events needs attention, as multiple local minima can occur within a single transit dip. There is also potential to

improve the system's performance by concentrating on a smaller number of relevant stars, considering more robust database options, among other improvements.

With potential project evolution, and the application of more sophisticated machine learning algorithms such as deep learning, TensorFlow and its Keras submodule could be considered potential tools for solving classification tasks in time-series data. These platforms offer advanced capabilities for building and training complex neural network models, which are essential for effectively analyzing and interpreting TESS telescope data. This is particularly relevant given that our data contains non-linear patterns, which traditional machine learning models often struggle to capture.

Moreover, there are various parameters for the algorithms used (such as wotan, moving averages, and linear regression window sizes, different wotan methods, transit width, and depth) that could be examined in greater detail. This examination would aim to identify the most suitable option or even to dynamically parameterize them based on the star type, noisiness, and other factors, rather than relying on static settings.

References

1. Pavlenko, Ya., Kulyk, I., Shubina, O., Vasylenko, M., Dobrycheva, D., & Korsun, P. (2022). **New exocomets of β Pic.** Astronomy & Astrophysics, 660, A49. https://doi.org/10.1051/0004-6361/202142111
2. Hartogh, P., Lis, D., Bockelée-Morvan, D. et al. **Ocean-like water in the Jupiter-family comet 103P/Hartley 2.** Nature 478, 218–220 (2011). https://doi.org/10.1038/nature10519
3. Pence, W. D., Chiappetti, L., Page, C. G., Shaw, R. A., & Stobie, E. (2010). **Definition of the Flexible Image Transport System (FITS), version 3.0**. Astronomy & Astrophysics, 524, A42. https://doi.org/10.1051/0004-6361/201015362
4. Exoplanet Exploration Program (n.d.). **Transit method of Exoplanet detection.** Retrieved from https://exoplanets.nasa.gov/alien-worlds/ways-to-find-a-planet/#/2
5. The National Aeronautics and Space Administration (n.d.) **Kepler / K2 mission.** Retrieved from https://science.nasa.gov/mission/kepler
6. TESS Science Team. (n.d.). **Transiting Exoplanet Survey Satellite (TESS)**. Massachusetts Institute of Technology. Retrieved 30.10.2023, from https://tess.mit.edu/
7. Zieba, S., Zwintz, K., Kenworthy, M. A., & Kennedy, G. M. (2019). **Transiting exocomets detected in broadband light by tess in the β pictoris system**. Astronomy & Astrophysics, 625, L13. https://doi.org/10.1051/0004-6361/201935552
8. Kennedy, G. M., Hope, G., Hodgkin, S. T., & Wyatt, M. C. (2018). **An automated search for transiting exocomets.** Monthly Notices of the Royal

Astronomical Society, 482(4), 5587-5596. https://doi.org/10.1093/mnras/sty3049

9. Rappaport S., Vanderburg A., Jacobs T., et. al. (February 2018) **Likely transiting exocomets detected by Kepler**, Monthly Notices of the Royal Astronomical Society, Volume 474, Issue 2, February 2018, Pages 1453–1468, https://doi.org/10.1093/mnras/stx2735
10. Michael Hippke, Trevor J. David, Gijs D. Mulders, and René Heller (2019, September 11). **Wōtan: Comprehensive Time-series Detrending in Python.** Astron. J. 158, 143. https://iopscience.iop.org/article/10.3847/1538-3881/ab3984/pdf
11. Project Jupyter. (n.d.). **Project Jupyter.** Retrieved from https://jupyter.org/
12. Plotly team (n.d.). **Dash by Plotly.** Retrieved from https://dash.plotly.com/
13. Scikit-learn developers (BSD License) (n.d.) **Scikit-learn.** Retrieved from https://scikit-learn.org/stable/
14. Brown, S. (n.d.). **The C4 Model for Visualizing Software Architecture.** Retrieved from https://c4model.com/
15. Structurizr. (n.d.). **Structurizr: Software architecture models as code.** Retrieved from https://structurizr.com/
16. The Mikulski Archive for Space Telescopes. Transiting Exoplanet Survey Satellite (TESS). (n.d.). **TESS: All-sky transit survey**. Retrieved from https://archive.stsci.edu/tess
17. Centre de Données astronomiques de Strasbourg (n.d.). **SIMBAD Astronomical Database.** Retrieved from https://simbad.u-strasbg.fr/simbad/
18. Segner, M. (2023, June 14). **Data Pipeline Architecture Explained: 6 Diagrams and Best Practices.** Monte Carlo Data. Retrieved from https://www.montecarlodata.com/blog-data-pipeline-architecture-explained/

19. DB diagram. (n.d.). dbdiagram.io: **Database Relationship Diagrams Design Tool**. Retrieved from https://dbdiagram.io/

20. High Energy Astrophysics Science Archive Research Center (HEASARC), NASA. (n.d.). **TESS Data Products.** Retrieved October 30, 2023, from https://heasarc.gsfc.nasa.gov/docs/tess/data-products.html

21. Wang, Z. (2023). **Extrasolar Planet Candidates Identified by Single Transit from TESS.** Journal of Physics: Conference Series, 2441, 012030. https://iopscience.iop.org/article/10.1088/1742-6596/2441/1/012030/pdf

22. Lightkurve Collaboration, Open-source, 2018. **Lightkurve**. Retrieved from https://docs.lightkurve.org/

23. Michael Hippke (n.d.). **Wotan, Detrending algorithms.** Retrieved from https://pypi.org/project/wotan/

Acknowledgments

This project could not be done without the work of the scientists from Main Astronomical Observatory of National Academy of Sciences of Ukraine. In particular: Iryna Kulyk, Daria Dobrycheva, Yakiv Pavlenko, Maksym Vasylenko, Olena Shubina, Igor Lukyanyk, and thanks to Olena Kopaniiets for introducing us to each other.

This research made use of Lightkurve, a Python package for Kepler and TESS data analysis (Lightkurve Collaboration, 2018), Dash by Plotly corporation, pandas and numpy.

Images and logo are generated with DALL-E model.