Detection of Unusual Patterns in Stellar Light Curves to Identify Potential Alien Megastructures Using Machine Learning

Sara Hammouch Benabdellah

28 de juny de 2025

Resum– Aquest treball desenvolupa una pipeline no supervisada per detectar anomalies en corbes de llum de Kepler. S'hi aplica preprocesament, reducció dimensional amb PCA i dos models d'aprenentatge automàtic: Isolation Forest i un Autoencoder profund. Els resultats demostren la seva eficàcia en la detecció d'estrelles anòmales com l'Estrella de Tabby. El sistema facilita l'anàlisi massiva de dades astronòmiques de forma eficient.

Paraules clau– corbes de llum estel·lars, detecció d'anomalies, aprenentatge no supervisat, Isolation Forest, Autoencoder, Anàlisi de Components Principals, Kepler, error de reconstrucció, variabilitat fotomètrica, valors atípics, megastructures artificials, esfera de Dyson, Estrella de Tabby, astroinformàtica.

Abstract– This project builds an unsupervised pipeline to detect anomalies in Kepler light curves. It combines preprocessing, PCA, and two machine learning models: Isolation Forest and a deep Autoencoder. Results confirm the method's effectiveness at spotting unusual stars like Tabby's Star. The system enables efficient large-scale analysis of astronomical data.

Keywords– stellar light curves, anomaly detection, unsupervised learning, Isolation Forest, Autoencoder, Principal Component Analysis, Kepler, reconstruction error, photometric variability, outliers, artificial megastructures, Dyson sphere, Tabby's Star, astroinformatics.

1 Introduction – Context of the Work

The detection of anomalies in large datasets has become a fundamental challenge in various scientific and technological fields. In astrophysics, a particularly fascinating application involves analyzing stellar light curves—time series representing the brightness of stars—to identify deviations from expected stellar behavior. These anomalies may stem from rare natural phenomena, observational noise, or even large-scale artificial constructs, such as the hypothetical alien megastructures discussed in recent scientific literature [1].

This bachelor's thesis investigates the use of unsupervised machine learning techniques to detect unusual patterns in stellar light curves obtained from space telescopes like *Kepler* [2] and *TESS* [3].

• Contact e-mail: 1638922@uab.cat
• Specialization: Computer Engineering
• Supervised by: [Yolanda Benítez Fernández] ([Area of Computer Science and Artificial Intelligence])
• Academic year: 2024/25

These missions have provided astronomers with vast quantities of photometric data, enabling the application of data-driven approaches for anomaly detection and scientific discovery.

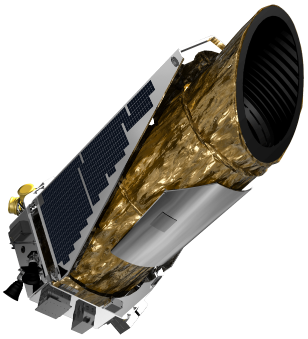

Figura 1: Representation of the *Kepler* space telescope, which collected the light curves used in this study.

Technically, the project follows a structured workflow: light curves are first preprocessed—cleaned, interpolated, and normalized—before applying dimensionality reduction using Principal Component Analysis (PCA). Then, two unsupervised algorithms — Isolation Forest and Autoencoder — are employed to identify outlier patterns. These may point to interesting astrophysical phenomena or non-standard behaviors worthy of further investigation.

This document is organized as follows. Section 2: Objectives presents the goals of the study. Section 3: State of the Art provides an overview of related research and methodologies. Section 4: Methodology details the overall approach adopted in this work. Section 5: Analysis and Design explains the motivations behind the architecture, design choices, and pipeline components. Section 6: Development describes the implementation of the system. Section 7: Results presents and discusses the outcomes of the anomaly detection process. Section 8: Conclusions summarizes the findings. Finally, Section 9: Future Work outlines potential directions for extending and improving the system in future research.

With this work, we aim to contribute a semi-automated framework for detecting anomalies in stellar photometric data, supporting modern astrophysical research and encouraging the exploration of unconventional scientific hypotheses.

2 OBJECTIVES

This section presents both the general objective of the project and a breakdown into specific goals that guide the development and implementation of the work. The objectives reflect both the scientific purpose of the study and the academic learning it aims to support.

2.1 General Objective

The main objective of this project is to develop a methodology capable of detecting and characterizing anomalies in stellar light curves obtained from the *Kepler* space missions. These anomalies may be linked to unknown astrophysical phenomena, unusual stellar behaviors, or, in rare cases, to hypothetical signatures of artificial megastructures such as Dyson spheres.

In addition to identifying unexpected variations in stellar brightness, the project also aims to explore the limitations of traditional data analysis methods and propose modern strategies based on unsupervised machine learning. This includes the use of statistical tools and neural models for effective pattern recognition within large astronomical datasets.

2.2 Specific Objectives

The following list outlines the specific objectives of the project:

- **To understand the nature of anomalies in light curves** and define criteria to distinguish truly unusual variations from noise or known behaviors.
- **To explore and compare different anomaly detection strategies**, including statistical techniques and machine learning models such as Isolation Forests and Autoencoders.
- **To analyze and interpret detected anomalies** in the context of existing astrophysical knowledge and documented stellar behaviors.
- **To develop technical and methodological skills** in data preprocessing, dimensionality reduction, and unsupervised learning, applicable to large-scale astronomical datasets.

For further reference, a set of functional and non-functional system requirements is provided in *Annex B*.

3 State of the Art

The analysis of stellar light curves has played a crucial role in the detection of exoplanets and other astrophysical phenomena. Thanks to space missions such as *Kepler* [2] and *TESS* [3], the availability of large-scale photometric datasets has enabled the identification of anomalous variations in stellar brightness, some of which may be related to yet unexplained phenomena.

One of the most intriguing cases is **KIC 8462852**, also known as *Tabby's Star*, which **exhibited** irregular and deep dips in brightness without a clear explanation [1]. Hypotheses range from large clouds of comets to unknown astrophysical processes, and even—more speculatively—to artificial structures such as **Dyson spheres** [4]

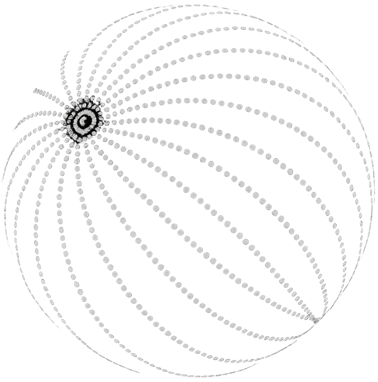

Figura 2: Artistic representation of a Dyson sphere—an artificial megastructure theorized to enclose a star and harness its energy, potentially altering the observed light curve.

Several research efforts have been dedicated to identifying stars with similar anomalous patterns using both statistical techniques and large-scale data analysis. For instance, **Principal Component Analysis (PCA)** has been used to detect outliers in light curve databases, while machine learning models have been applied to automate the detection process.

The application of **machine learning** in astronomy has gained momentum in recent years due to its capacity to extract complex patterns from massive datasets. Deep learning methods have been applied in various domains, including galaxy classification and the analysis of time series such as light curves [5].

In the context of anomaly detection, three main categories of approaches have been explored:

- **Statistical methods**, such as PCA, which help reduce dimensionality and highlight deviations from expected light curve behaviors.
- **Anomaly detection models**, including Isolation Forest, which have been effective at flagging outliers across large datasets.
- **Deep learning models**, particularly Autoencoders and LSTMs, which can capture nonlinear dependencies and complex temporal patterns, enabling the identification of subtle or rare anomalies.

Despite notable progress, important challenges remain. Differentiating between genuine astrophysical anomalies and instrumental artifacts is critical. Additionally, interpretability continues to be a limitation, especially when using black-box models such as deep neural networks.

As data volumes continue to grow and algorithms improve, the field is moving closer to detecting more complex anomalies. These efforts may eventually lead to the discovery of new classes of astrophysical phenomena or even provide indirect evidence for artificial structures in space.

4 Methodology

The methodology followed in this project has been based on an iterative and flexible workflow, allowing progressive improvements and adaptation throughout the development process. Rather than following a fixed plan, the tasks evolved as new challenges and insights appeared, particularly during data preprocessing and model tuning.

Initially, GitHub was used for version control and local development was carried out in Python using PyCharm. However, due to memory limitations on the local machine—especially when working with large volumes of stellar data and training models—the entire workflow was eventually migrated to a cloud-based environment using Google Drive and Google Colab. An attempt was made to connect via SSH to a remote server, but it was ultimately discarded due to configuration difficulties.

All documentation was written in Overleaf, which allowed centralized and organized access to the report from any device. Project planning and coordination were supported by periodic meetings with the supervisor, and tasks were progressively defined and adjusted as results emerged.

A visual summary of the project timeline is included in *Annex A*, where a Gantt chart outlines the structure of the main development phases.

5 Analysis and Design

This section outlines the rationale behind the design choices and structure of the anomaly detection system for stellar light curves. The goal was to build a modular, extensible, and interpretable pipeline capable of identifying unusual photometric behaviors without relying on labeled data.

5.1 Problem Analysis

The input to the system consists of raw light curve data from space telescopes such as *Kepler*. These light curves exhibit high dimensionality, irregular sampling, and are often affected by noise and instrumental artifacts. The main challenge is to detect anomalies—patterns that deviate from typical stellar behavior—without prior examples or classifications.

Traditional supervised learning approaches are unsuitable in this context due to the scarcity of labeled anomalous cases. Therefore, an unsupervised strategy was selected to detect outliers based on reconstruction errors and distributional deviations in a reduced feature space.

5.2 System Architecture

The system is structured as a sequential pipeline with the following components:

- **Preprocessing**: Raw CSV files are loaded and normalized to remove magnitude offsets and facilitate comparison across stars.
- **Dimensionality Reduction (PCA)**: Principal Component Analysis is applied to extract dominant trends in the light curves and reduce dimensionality while preserving variance.
- **Anomaly Detection Models**:
  - An *Autoencoder* is trained to reconstruct light curves. Large reconstruction errors are interpreted as potential anomalies.
  - An *Isolation Forest* is applied on the PCA-reduced data to identify spatial outliers in the feature space.
- **Post-processing and Evaluation**: The outputs of both models are compared to identify consistently flagged stars, and results are visualized through PCA projections and tables.

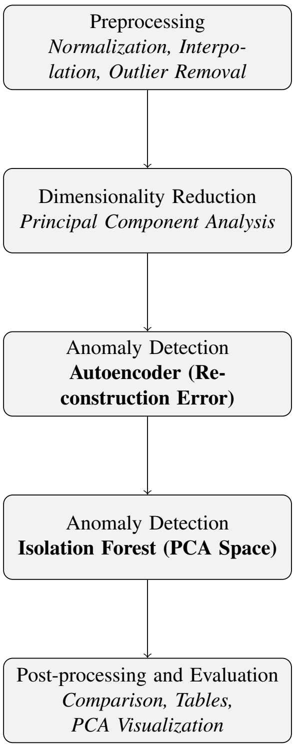
Preprocessing
Normalization, Interpolation, Outlier Removal
Dimensionality Reduction
Principal Component Analysis
Anomaly Detection
Autoencoder (Reconstruction Error)
Anomaly Detection
Isolation Forest (PCA Space)
Post-processing and Evaluation
Comparison, Tables, PCA Visualization

Figura 3: Vertical architecture of the anomaly detection pipeline.

5.3 Design Decisions

The design phase is a critical part of any data-driven system, as it directly impacts both the robustness of the implementation and the scientific value of the results. In this project, several architectural and methodological choices were made to balance computational efficiency, interpretability, and extensibility. These decisions were based not only on the characteristics of the data, but also on the practical constraints of the development environment.

- **Use of PCA before Isolation Forest**: Principal Component Analysis was applied as a dimensionality reduction step to enhance the performance of Isolation Forest, which performs better in low-dimensional spaces. Alternatives such as t-SNE or UMAP were considered, but PCA was preferred due to its speed, linearity, and interpretability in anomaly detection scenarios.

- **Unsupervised Deep Learning**: An Autoencoder was selected for its ability to reconstruct temporal structures and capture nonlinear relationships in light curve data. Although other models like LSTM Autoencoders were explored, the classic feedforward architecture provided a good trade-off between complexity and performance, and was easier to train on limited hardware.
- **Google Colab Integration**: The use of Google Colab, combined with structured storage on Google Drive, ensured reproducibility and accessibility. A local server setup was initially attempted via SSH, but discarded due to configuration complexity and limited memory.
- **Modular Structure**: The codebase was organized into modular scripts with clear separation of concerns—data loading, preprocessing, model training, anomaly detection, and visualization. This modularity simplifies debugging and facilitates future integration of new algorithms.

```
project_root/
|- src/
|   |- preprocessings.py/
|   |- pca_analysis.py/
|   |- autoencoder_model.py/
|- notebooks/
|   |- download_data.ipynb
|- scripts/
|   |- run_autoencoder.py
|   |- run_pca.py
|   |- run_isolation_forest.py
|   |- run_comparison.py
|- data/
|   |- raw/
|   |- processed/
|       |- train/
|       |- test/
|- results/
```

Figura 4: Directory structure of the project showing source code, scripts, data organization, and notebooks.

6 Development

6.1 Downloading

A total of **169 light curves** from the *Kepler* mission were downloaded using the `lightkurve` library [6]. The selected stars were chosen to avoid overlap with previously analyzed targets, aiming for a representative and diverse sample.

The script was executed in a Colab environment, with all data stored in Google Drive. For each star, all available segments were stitched into a single time series. The resulting files were saved as CSV in the `data/raw/` directory, with a consistent naming format such as `curva_luz_KIC_XXXXXXX.csv` or `curva_luz_Kepler-YY.csv`.

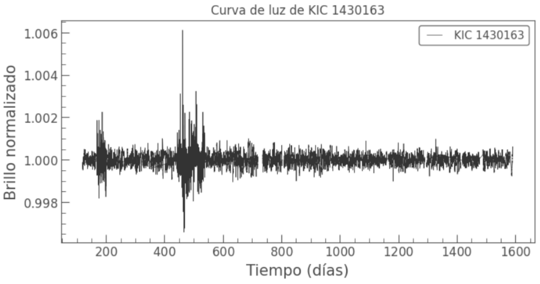

Figura 5: Example of a light curve downloaded and stitched using `lightkurve`.

6.2 Preprocessing

Before applying any machine learning technique, the raw light curves from the *Kepler* mission [2] were preprocessed to ensure data quality and consistency. This step was essential to remove observational artifacts and prepare the data for dimensionality reduction and anomaly detection.

First, each light curve was loaded from its corresponding CSV file and normalized to remove amplitude biases. Outliers were then identified and removed using a rolling window approach: any data point that deviated more than 2σ from the local mean was considered an outlier and excluded. Finally, gaps in the data were filled via linear interpolation to obtain continuous time series of uniform length.

The resulting clean light curves were split into two subsets. The first subset, referred to as the training set, was used to fit the PCA model, train the Isolation Forest, and learn the Autoencoder. The second subset was reserved for testing and evaluation. This test set includes both normal and potentially anomalous stars, allowing for a robust assessment of the detection models under realistic conditions.

6.3 Dimensionality Reduction with PCA

To reduce the dimensionality of the time series and better visualize potential anomalies, a **Principal Component Analysis (PCA)** was applied to the preprocessed light curves. Each curve, resampled to a fixed length and standardized, was projected onto the first two principal components. This projection captures the directions of maximum variance in the dataset.

The PCA was applied independently to the `train` and `test` datasets, using the same function pipeline. Figures 6 and 7 show the PCA results for each split.

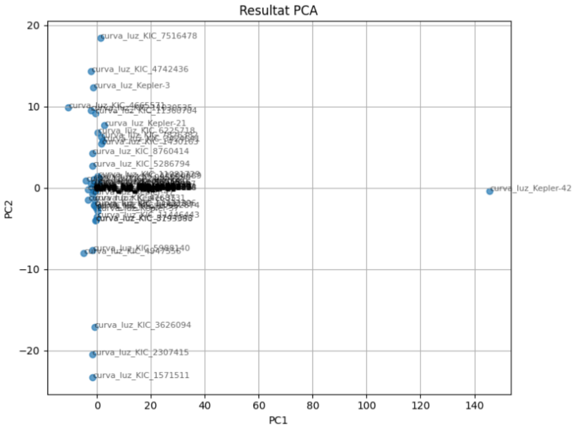

Figura 6: PCA projection of the `train` set. Most light curves form a dense cluster, but several lie farther away from the core region.

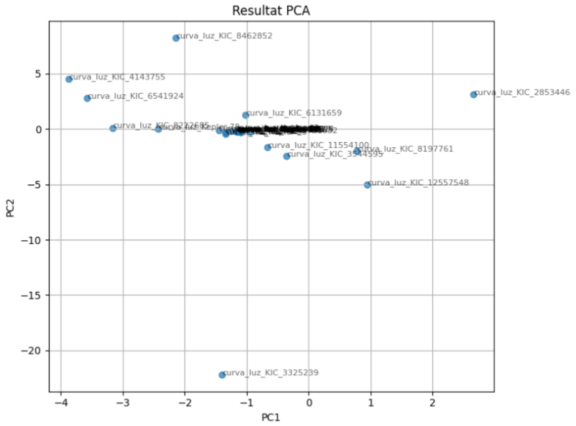

Figura 7: PCA projection of the `test` set. A few light curves are notably distant, suggesting anomalous variability.

While most curves are grouped in a compact region, some are positioned further away in the PC1–PC2 plane. These dispersed points often correspond to known anomalous stars, such as *KIC 8462852* or *KIC 12557548*, confirming that PCA already provides a preliminary indication of outlier behavior before any machine learning model is applied.

This transformation was essential for subsequent anomaly detection with Isolation Forest, which operates directly on the PCA-reduced space.

6.4 Anomaly Detection with Isolation Forest

To detect unusual patterns in stellar brightness, an **Isolation Forest** model was trained on the training set projected onto the first two principal components. The model was then applied to the test set, producing an anomaly score and a binary label for each light curve: `+1` for normal and `−1` for anomalous.

Figure 8 displays the anomaly detection results, where each point represents a light curve in the PCA space. The color indicates the predicted class. Most stars form a compact central region, but several outliers are clearly visible.

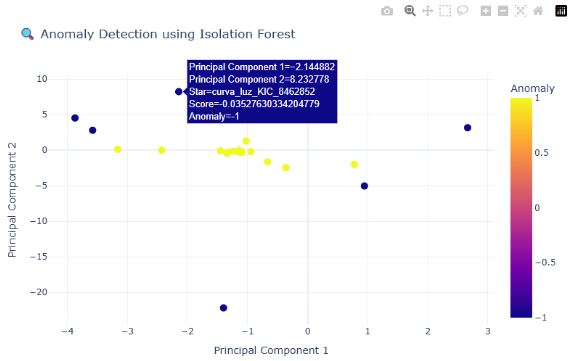

Figura 8: Anomaly detection using Isolation Forest on PCA-transformed light curves. Blue points represent detected anomalies (−1), while yellow points are considered normal (+1).

In this example, hovering over a point reveals detailed metadata about the corresponding star, including its ID, coordinates in PCA space, anomaly score, and predicted label.

6.5 Anomaly Detection with Autoencoder

To complement the PCA and Isolation Forest pipeline, a deep **Autoencoder** was implemented to detect anomalous light curves based on their reconstruction error. The model was trained exclusively on the training set, which contains light curves assumed to be normal. Its objective was to learn a compact representation capable of accurately reconstructing the input data.

The Autoencoder architecture consisted of three encoding layers with ReLU activations and L2 regularization, and a symmetric decoder. Inputs were first normalized using `MinMaxScaler`, and the output layer used a sigmoid activation to ensure values remained within the $[0, 1]$ range. The model was trained using the mean squared error (MSE) loss function:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2$$

Once trained, the Autoencoder was applied to the test set. The reconstruction error was computed for each curve, and anomalies were identified by comparing this error against a threshold. This threshold was automatically estimated using the *KneeLocator* method [7]. If no clear inflection point was found, the 95th percentile of training errors was used as a fallback. Figure 9 illustrates how the Autoencoder behaves on the training set. The majority of curves are reconstructed with minimal error, indicating that the model has correctly learned the underlying data distribution. The chosen threshold, visible in both plots, effectively separates normal from potentially anomalous patterns.

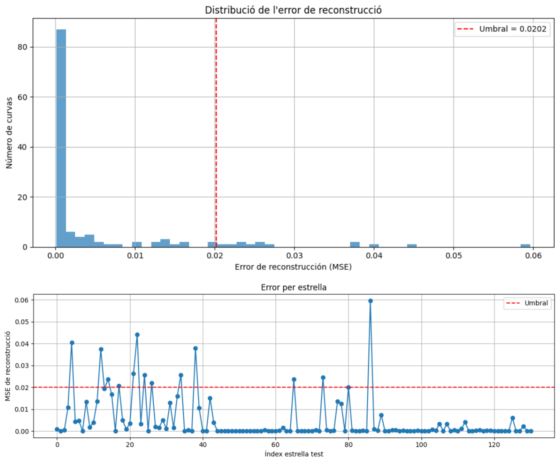

Figura 9: Reconstruction error on the training set. Top: histogram showing the distribution of MSE across all curves. Bottom: per-star reconstruction error, with the anomaly threshold indicated by the red dashed line.

6.6 Comparison of Results

To better understand the behavior of both models, Table 2 summarizes a subset of stars and the predictions made by the Autoencoder and Isolation Forest. The values −1 and 1 indicate anomaly and normal classification, respectively.

Star	**Autoencoder**	**Isolation Forest**
KIC 12557548	-1	-1
KIC 8462852	-1	-1
KIC 3544595	-1	1
Kepler-78	-1	1
KIC 3861595	1	1
Kepler-55	-1	1
Kepler-10	1	1
KIC 3325239	-1	-1
KIC 4143755	-1	-1
KIC 6131659	-1	1

Taula 1: Anomaly detection results. −1 indicates a detected anomaly.

The comparison shows that both models agreed on several known anomalous stars, including **KIC 8462852** and **KIC 12557548**. However, the Autoencoder identified a larger number of outliers overall, flagging stars such as **Kepler-55** and **KIC 6131659**, which the Isolation Forest did not.

This difference highlights the complementary nature of both approaches. While the Autoencoder learns to capture fine reconstruction deviations from typical light curves, the Isolation Forest focuses on outliers in the low-dimensional PCA space. Using both methods in parallel improves robustness and confidence in anomaly detection.

6.7 Testing and Validation

To ensure the reliability of the implemented methods, a series of validation tests were performed across all stages of the pipeline. These tests aimed to verify both the internal consistency of the functions and the expected behavior of the models under controlled conditions.

Preprocessing validation: A functional test was designed to verify the integrity of the preprocessing step. Several raw light curves were processed and compared before and after applying the outlier removal method. Visual inspection confirmed that isolated noise points were effectively eliminated, while the overall shape of the signal was preserved. Additionally, a verification routine checked that the output format matched the expected structure `(time, normalized_flux)` and that files were correctly saved.

PCA validation with synthetic data: To validate the PCA transformation, synthetic datasets were constructed, including curves with normal and deliberately anomalous patterns. The PCA projection correctly separated the anomalous samples in the reduced space, confirming its ability to preserve meaningful variance. Moreover, in real test data, known anomalies such as *KIC 8462852* were positioned far from the main cluster.

Isolation Forest behavior: A unit test was designed where the model was trained on tightly clustered normal data and evaluated on test data containing injected anomalies. The algorithm successfully flagged the outliers while leaving normal samples unaffected, validating its effectiveness. Additionally, in the real dataset, several known anomalous stars were consistently detected, reinforcing confidence in the method's performance.

Autoencoder test with known outputs: A controlled experiment was conducted using synthetic curves, where the Autoencoder was trained on sinusoidal-like normal curves and tested on corrupted or flat signals. The reconstruction error was significantly higher for the anomalous samples, and the model correctly classified them based on a threshold estimated using the *KneeLocator*. Furthermore, real examples confirmed that known anomalies like *KIC 12557548* produced high reconstruction errors, validating the detection process.

Conclusion on validation: The test cases demonstrated that each module of the pipeline—preprocessing, dimensionality reduction, and anomaly detection—behaves as expected both in synthetic scenarios and on real astrophysical data. This confirms that the methodology is robust and reliable for the detection of unusual light curves.

7 RESULTS

The anomaly detection models were applied to a test set of 36 preprocessed light curves. As shown in Figure 10, the distribution of reconstruction errors from the Autoencoder revealed a small group of clear outliers. The anomaly threshold—estimated automatically—allowed us to isolate these cases with minimal ambiguity.

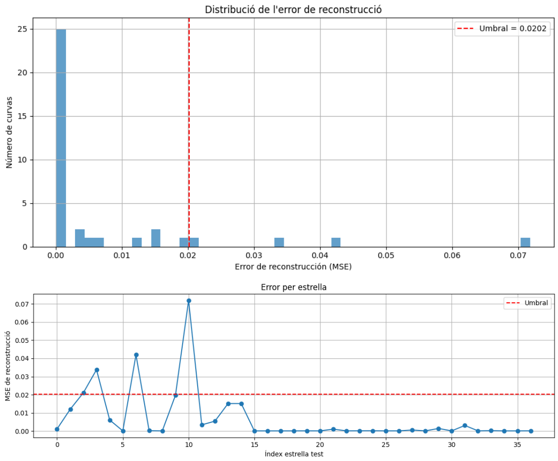

Figura 10: Reconstruction error on the test set. Top: histogram of MSE values. Bottom: per-star reconstruction error, with the red dashed line marking the anomaly threshold.

In total, the Autoencoder flagged 10 stars and the Isolation Forest identified 7, with 6 overlaps. Table 2 highlights a subset of stars and their classification by both models. Known anomalies such as **KIC 8462852** and **KIC 12557548** were successfully detected by both, confirming the reliability of the approach.

Star	Autoencoder	Isolation Forest
KIC 12557548	-1	-1
KIC 8462852	-1	-1
KIC 3544595	-1	1
Kepler-78	-1	1
KIC 3861595	1	1
Kepler-55	-1	1
Kepler-10	1	1
KIC 3325239	-1	-1
KIC 4143755	-1	-1
KIC 6131659	-1	1

Taula 2: Anomaly detection results. `-1` indicates a detected anomaly.

While some cases were detected only by one model, the overlap between methods enhances confidence in those findings. The Autoencoder, in particular, was able to detect more subtle deviations not always captured in the PCA space.

From a scientific perspective, the anomalous stars include both known and potentially new candidates of interest. Some show irregular variability patterns already discussed in the literature, while others—flagged solely by one model—may warrant follow-up analysis to determine if their behavior is due to astrophysical phenomena, instrumental noise, or something less conventional.

8 Conclusions

This work has developed an unsupervised pipeline capable of detecting anomalous stellar light curves using publicly available data from the *Kepler* mission [2]. By combining Principal Component Analysis, Isolation Forest, and Autoencoders, we designed a methodology that successfully identifies atypical photometric behaviors without requiring labeled examples.

The approach was validated on a dataset of 169 stars, and the results confirmed the system's ability to recover known anomalous cases—most notably **KIC 8462852** and **KIC 12557548**—as well as to surface other candidates worthy of astrophysical attention. The complementary nature of the two anomaly detection models proved beneficial: the Autoencoder was effective in capturing subtle deviations in light curve reconstruction, while the Isolation Forest identified spatial outliers in reduced feature space.

Beyond detecting already studied stars, the methodology also flagged objects not yet widely characterized, such as **Kepler-55** and **KIC 6131659**, suggesting its potential for uncovering new classes of stellar variability or instrumental artifacts.

Ultimately, being able to systematically detect "what does not fit" in stellar light curves opens a pathway to discovery. In a universe where most objects behave predictably, it is the exceptions that may one day lead to the most surprising insights.

9 Future Work

The current pipeline opens several promising directions for future development. First, applying the methodology to larger datasets, such as those from the *TESS* mission, could enable the detection of additional anomalies across different time scales and stellar populations.

Second, the integration of temporal deep learning models—like LSTM networks or Transformer-based architectures—could improve the detection of dynamic or periodic behaviors that static projections may overlook. This would be particularly useful for identifying evolving anomalies or transient events.

Third, enhancing model interpretability and incorporating domain-specific astrophysical knowledge could help distinguish between true physical phenomena and artifacts or noise. This may involve coupling the anomaly detection outputs with stellar classification tools or follow-up observational strategies.

Finally, while speculative, the ability to detect subtle, unexplained deviations in light curves could one day contribute to the search for unconventional astrophysical scenarios, such as artificial megastructures or other non-natural dimming patterns. Establishing rigorous, data-driven baselines now may be key to recognizing such signatures if they ever appear.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Yolanda Benítez Fernández, for her dedication, constant support, and for guiding this work with enthusiasm and commitment throughout all its stages. Her encouragement has been essential in helping me stay motivated and move forward with confidence.

I would also like to thank Jordi Pons Aróztegui, coordinator of the Bachelor's Degree in Computer Engineering, for always being there throughout my studies.

To my family, thank you for always believing in me, for standing by my side, and for offering your unconditional help and support. This journey would have been much harder without you.

To all of you, thank you from the bottom of my heart.

Referències

[1] T. S. Boyajian and et al., "Planet hunters ix. kic 8462852 – where's the flux?" *Monthly Notices of the Royal Astronomical Society*, vol. 457, no. 4, pp. 3988–4004, 2016.

[2] W. J. Borucki *et al.*, "Kepler planet-detection mission: Introduction and first results," *Science*, vol. 327, no. 5968, pp. 977–980, 2010.

[3] G. R. Ricker *et al.*, "Transiting exoplanet survey satellite (tess)," *Journal of Astronomical Telescopes, Instruments, and Systems*, vol. 1, no. 1, p. 014003, 2015.

[4] F. J. Dyson, "Search for artificial stellar sources of infrared radiation," *Science*, vol. 131, no. 3414, pp. 1667–1668, 1960.

[5] D. Baron, "Machine learning in astronomy: a practical overview," *Frontiers in Astronomy and Space Sciences*, vol. 6, p. 57, 2020.

[6] G. Barentsen and et al., "Lightkurve: Kepler and tess time series analysis in python," *Astrophysics Source Code Library*, 2019, ascl:1812.013. [Online]. Available: https://doi.org/10.5281/zenodo.1181928

[7] V. Satopää, J. Albrecht, D. Irwin, B. Raghavan, and C. Spence, "Finding a "kneedleïn a haystack: Detecting knee points in system behavior," *2011 31st International Conference on Distributed Computing Systems Workshops*, p. 166–171, 2011.

[8] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[9] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019.

[10] I. T. Jolliffe and J. Cadima, *Principal component analysis*. Springer, 2016.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[12] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[13] Y. Zhang, Y. Zhao, and Y. Liu, "Deep learning based classification of light curves in astronomy," *Astrophysics and Space Science*, vol. 364, no. 5, pp. 1–11, 2019.

[14] I. Nun and et al., "Fats: Feature analysis for time series," *Astrophysics Source Code Library*, 2015, ascl:1506.001. [Online]. Available: https://ascl.net/1506.001

[15] J. T. VanderPlas, "Understanding the lomb–scargle periodogram," *arXiv preprint arXiv:1207.5578*, 2012.

[16] C. C. Aggarwal, "Outlier detection for temporal data: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 11, pp. 2319–2339, 2017.

[17] M. M. Ribeiro, S. Matos, and G. Bianconi, "Survey on outlier detection in time series," *ACM Computing Surveys*, vol. 53, no. 3, pp. 1–33, 2020.

[18] K. Hundman, V. Constantinou, H. Laporte, I. Colwell, and N. Soderstrom, "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding," *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, pp. 387–395, 2018.

[19] Y. Zhao, Z. Nasrullah, and Z. Li, "Pyod: A python toolbox for scalable outlier detection," *Journal of Machine Learning Research*, vol. 20, no. 96, pp. 1–7, 2019. [Online]. Available: http://jmlr.org/papers/v20/19-011.html

[20] M. Abadi, A. Agarwal, P. Barham, and et al., "Tensorflow: Large-scale machine learning on heterogeneous systems," https://www.tensorflow.org/, 2015, software available from tensorflow.org.

Annex A Project Timeline

Figure 11 shows the complete Gantt chart for the project, including the main tasks and their respective execution periods.

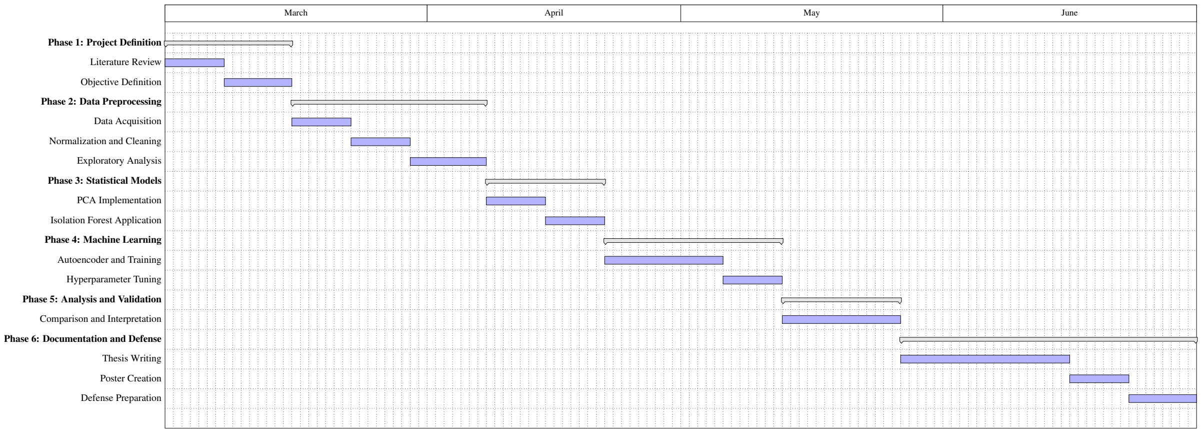

Figura 11: Detailed project planning (Gantt chart)

Annex B System Requirements

R-1: Import of Astronomical Data

Property	Description
Requirement ID	R-1
Title	Import of Astronomical Data
Description	The system must be able to automatically download data from the Kepler and TESS missions.
Priority	High
Type	Functional requirement
Dependencies	-

R-2: Preprocessing of Light Curves

Property	Description
Requirement ID	R-2
Title	Preprocessing of Light Curves
Description	The system must clean, normalize, and detect outliers in the light curves to prepare them for analysis.
Priority	High
Type	Functional requirement
Dependencies	R-1

R-3: Execution of Automated Tests

Property	Description
Requirement ID	R-3
Title	Execution of Automated Tests
Description	The system must include automated tests to validate the correct functioning of the different workflow stages.
Priority	High
Type	Non-functional requirement
Dependencies	R-2, R-5, R-6, R-7, R-8

R-4: Application of PCA

Property	Description
Requirement ID	R-4
Title	Application of PCA
Description	The system must apply Principal Component Analysis (PCA) to reduce the dimensionality of the data.
Priority	High
Type	Functional requirement
Dependencies	R-2

R-5: Application of Isolation Forest

Property	Description
Requirement ID	R-5
Title	Application of Isolation Forest
Description	The system must apply the Isolation Forest algorithm to detect anomalies in the light curves.
Priority	High
Type	Functional requirement
Dependencies	R-2, R-4

R-6: Development of Autoencoder

Property	Description
Requirement ID	R-6
Title	Development of Autoencoder
Description	The system must include an Autoencoder model for advanced anomaly detection in light curves.
Priority	High
Type	Functional requirement
Dependencies	R-2, R-4, R-5

R-7: Training of the Autoencoder Model

Property	Description
Requirement ID	R-7
Title	Training of the Autoencoder Model
Description	The system must allow training of the Autoencoder model on the preprocessed data.
Priority	High
Type	Functional requirement
Dependencies	R-2, R-6, R-14

R-8: Hyperparameter Tuning

Property	Description
Requirement ID	R-8
Title	Hyperparameter Tuning
Description	The system must facilitate the tuning of hyperparameters of the Autoencoder model.
Priority	High
Type	Functional requirement
Dependencies	R-6, R-7, R-15

R-9: Project Documentation

Property	Description
Requirement ID	R-9
Title	Project Documentation
Description	The system must generate graphical outputs and visual reports to document the workflow, the applied methodology, and the obtained results.
Priority	Medium
Type	Non-functional requirement
Dependencies	R-10, R-11

R-10: Data Visualization

Property	Description
Requirement ID	R-10
Title	Data Visualization
Description	The system must automatically generate plots showing the original curves, the detected outliers, and the final filtered curves.
Priority	Medium
Type	Functional requirement
Dependencies	R-2, R-9, R-11

R-11: Manual Validation of Results

Property	Description
Requirement ID	R-11
Title	Manual Validation of Results
Description	The system must allow visual validation of the detected anomalies to verify their astrophysical consistency.
Priority	Medium
Type	Functional requirement
Dependencies	R-10, R-9

R-12: Management of Large Files

Property	Description
Requirement ID	R-12
Title	Management of Large Files
Description	The system must handle large files properly, avoiding limitations such as those imposed by GitHub.
Priority	Low
Type	Non-functional requirement
Dependencies	R-1, R-2

R-13: Scalability and Execution Environment

Property	Description
Requirement ID	R-13
Title	Scalability and Execution Environment
Description	The system must be compatible with both local and cloud-based execution environments.
Priority	Medium
Type	Non-functional requirement
Dependencies	R-7, R-8

R-14: Dataset Selection and Partitioning

Property	Description
Requirement ID	R-14
Title	Dataset Selection and Partitioning
Description	The system must allow selecting and splitting the dataset into training and validation subsets.
Priority	High
Type	Functional requirement
Dependencies	R-7, R-15

R-15: Evaluation of Model Performance

Property	Description
Requirement ID	R-15
Title	Evaluation of Model Performance
Description	The system must allow evaluating the performance of the Autoencoder model using appropriate metrics.
Priority	High
Type	Functional requirement
Dependencies	R-7, R-8, R-14