Report

Nasa Space Apps Challenge 2025.

A World Away: Hunting for Exoplanets with AI.

Exoplanet Hunters

Team members:

Carolina Valdivia

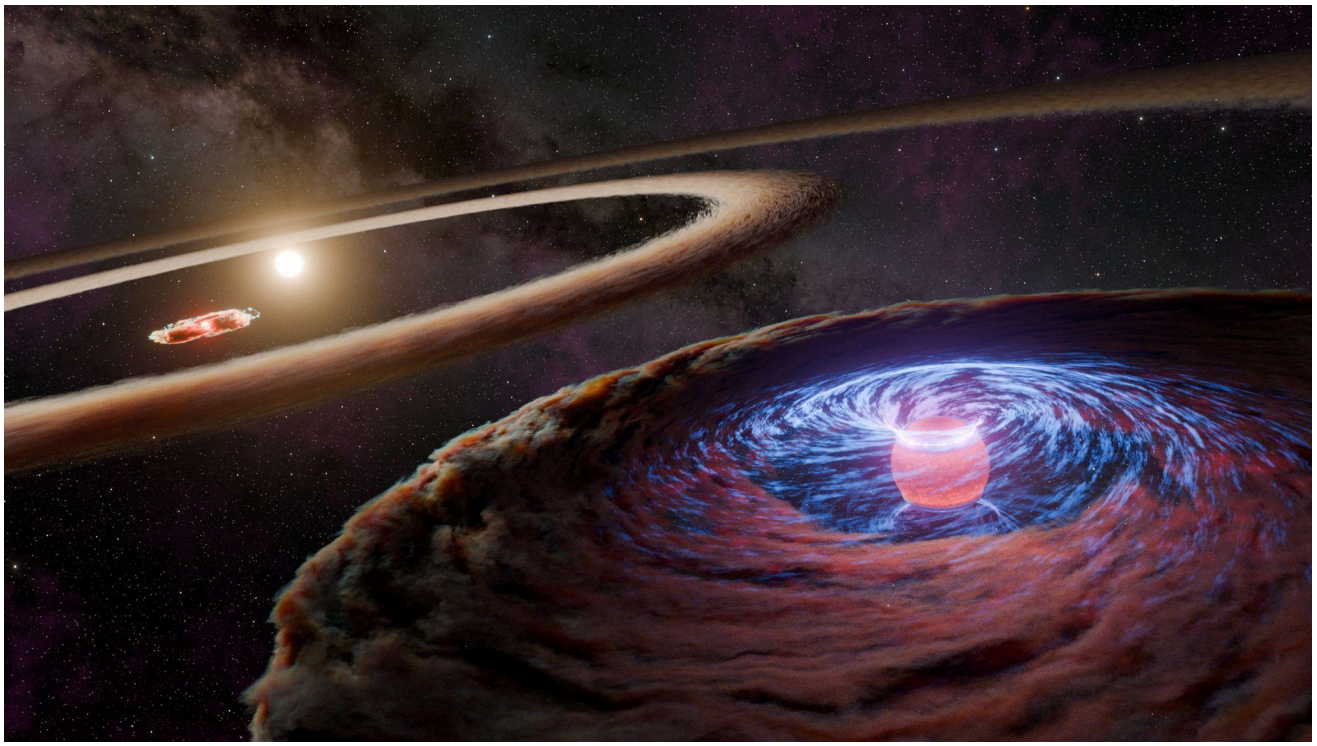
Diego Noriega

Emiliano Villalobos

Roberto Garcés

Saúl Perez

Ximena León



October 4th, 2025

Index

Introduction

In recent decades, the interest in discovering new worlds beyond our solar system has increased significantly, fueled by humanity's deep curiosity about the existence of other habitable planets. This pursuit has led to remarkable advancements in both space-based telescopes and detection techniques. Pioneering missions such as Kepler, its successor K2, and the ongoing Transiting Exoplanet Survey Satellite (TESS) have fundamentally enhanced our ability to observe and characterize exoplanets[1].

The most effective detection method used by these missions is the transit method, which identifies periodic dips in a star's brightness when a planet passes in front of it (figure 1). This approach has proven to be highly successful, confirming the existence of over 6,000 exoplanets. Despite these advances, most confirmed exoplanets are located within a relatively small region of our galaxy, with even the closest one, Proxima Centauri b, being 4 light-years away. Nonetheless, we now understand that there are more planets than stars in the Milky Way, making the search for exoplanets increasingly promising [2].

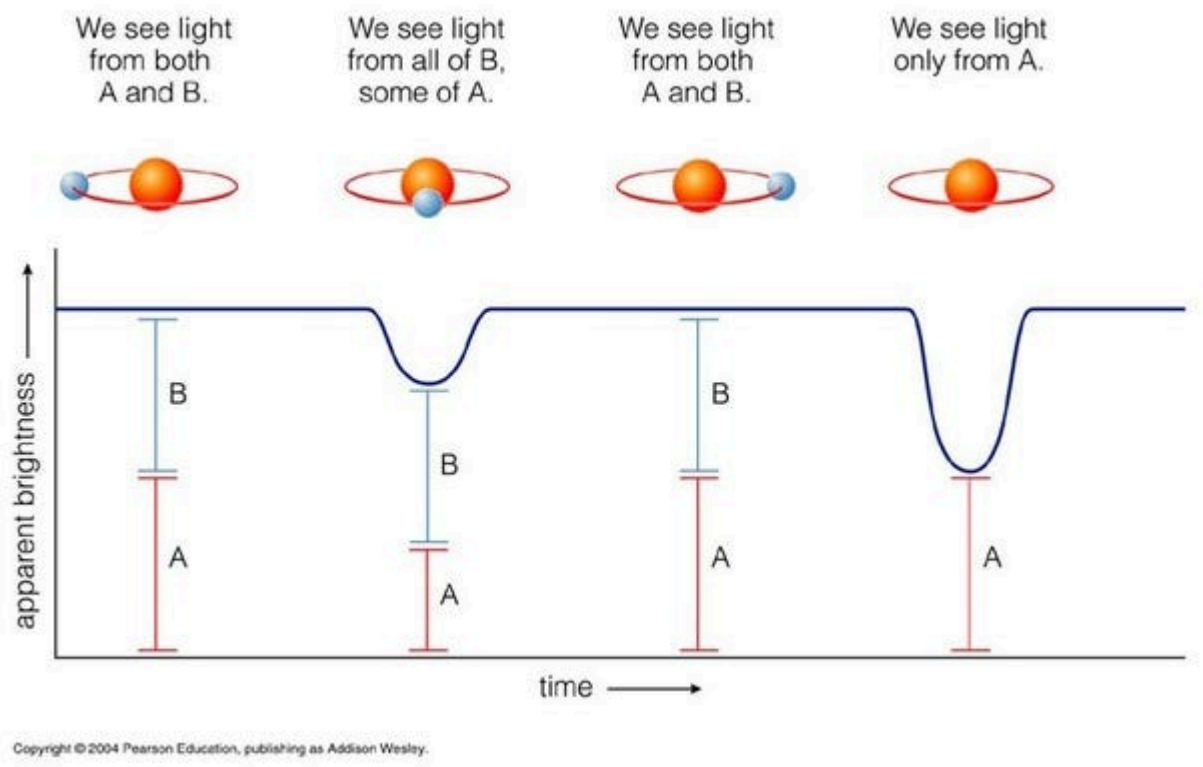


Figure 1. The transition model for obtaining light curves

The vast amount of photometric data generated by space telescopes like Kepler and TESS has exceeded our capacity for manual analysis. Traditionally, exoplanet detection required significant human effort to painstakingly analyze light curves, distinguishing true planetary signals from stellar noise or instrumental artifacts. However, the sheer scale of data from modern surveys has made manual classification time-consuming, inefficient, and susceptible to human bias[3].

To tackle these challenges, machine learning and deep learning have emerged as transformative tools. These techniques can automate the classification of transit signals, enabling rapid and accurate identification of exoplanet candidates. The objective is to maximize precision and recall identifying true exoplanets while minimizing false positives and thereby accelerating the discovery of potentially habitable worlds[3].

In this challenge, we aim to design and train a deep learning model capable of accurately classifying transit signals into three categories: Confirmed, Candidate, and False Positive. By utilizing NASA datasets and applying a machine learning model, we seek to replicate and

enhance the process traditionally undertaken by astronomers, paving the way for scalable and unbiased exoplanet discovery in the era of big data astronomy[4].

Justification

Extensive research on Machine Learning (ML) has identified a suitable framework for exoplanet detection using models such as Random Forest, k-Nearest Neighbors (KNN), Decision Tree, Logistic Regression, and Convolutional Neural Networks (CNNs) [5]. However, only Agnes *et al* (2023)[3] assesses the viability of Generative Adversarial Networks (GANs). We explored the possibility of using GANs as a suitable framework for exoplanet detection through transit photometry data, employing them as an auxiliary classifier designed to be effective in situations where imbalanced data is available for study.

For this purpose, we used NASA's *Kepler Objects of Interest* (KOI) database[6] due to its periodicity, as well as its high-utility targets and labels for classification tasks using GAN. Each KOI corresponds to Kepler targets with mission-standard products—such as long- and short-cadence light curves, target pixel files, and data-validation time series—available at MAST with bulk and API access. Therefore, the KOI dataset integrates seamlessly with our implementation, which leverages accessible open-source Python tools such as *Lightkurve* [7].

Method

Data Preprocessing

We start from Kepler long-cadence light curves fetched with Lightkurve using each target's KIC/kepid [6]. For every star we download and stitch quarters, then flatten to remove slow instrumental/trend systematics and remove outliers so single spikes don't dominate the learning signal. Next, we run a Box-Least-Squares (BLS) search on periods in [1, 20] days to estimate the best period, transit epoch (t_0), and duration; using these, we produce a folded representation of the series that concentrates the transit shape into phase space. The resulting 1-D sequence is then normalized to [-1, 1] to place all stars on a comparable scale, and finally padded or trimmed to a fixed length CURVE_LENGTH so the network can operate on uniform tensors of shape (L, 1).

Model Architecture

The model is a discriminator paired with a lightweight generator, as reported in Karimi *et al* (2025)[3] (figure 2). The generator ingests Gaussian noise, and uses a stack of Dense → BatchNorm → LeakyReLU(0.2) blocks to synthesize a 1-D sequence. A final tanh layer outputs values in [-1, 1].

The discriminator is a 1-D CNN: two Conv1D(32, kernel=7, same) blocks each followed by LeakyReLU and a stride-1 MaxPool1D to emphasize local patterns without aggressive downsampling. Features are flattened and passed through Dense(128) → Dense(64) with LeakyReLU. From this shared representation the network branches into two heads: (1) a sigmoid unit for real/fake (realness) using binary cross-entropy, and (2) a 3-way softmax for astrophysical class (0 = false positive, 1 = candidate, 2 = confirmed) trained with sparse categorical cross-entropy.

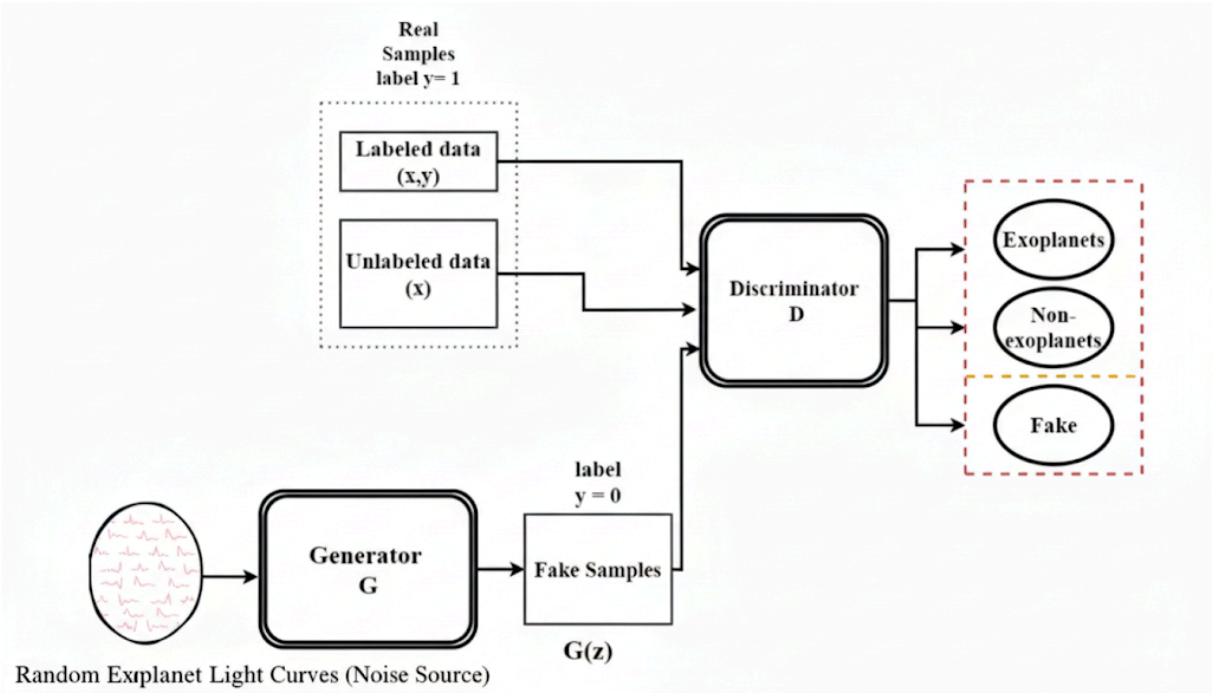


Figure 2. Model Architecture of GAN, reported in the literature as ExoSGAN.

Hyperparameters of the generator, discriminator and Training settings of the model are displayed on the table 1.

Generator

Operation	Feature Map / Units	Dropout	Non-linearity
Dense	800	0.0	LeakyReLU(0.2)

Dense	1600	0.0	LeakyReLU(0.2)
Dense	CURVE_LENGTH (=3197)	0.0	LeakyReLU(0.2)
Dense → Reshape	CURVE_LENGTH → (L,1)	0.0	**tanh**

Discriminator

Operation	**Kernel**	**Strides**	**Feature Map / Units**	**Non-linearity**
Conv1D	7×1	1×1	32	LeakyReLU(0.2)
Max-Pooling1D	2	**stride 1**	—	—
Conv1D	7×1	1×1	32	LeakyReLU(0.2)
Max-Pooling1D	2	**stride 1**	—	—
Dense	N/A	N/A	128	LeakyReLU(0.2)
Dense	N/A	N/A	64	LeakyReLU(0.2)
Output (realness)	N/A	N/A	1	**sigmoid**
Output (class)	N/A	N/A	3	**softmax**

Training

Setting	**Value**
Optimizer (both nets)	Adam (lr = **4e-5**, β_1 = **0.5**)
Batch size	**8**
Epochs	**150**

Latent dim (generator)	**100**
LeakyReLU slope	**0.2**
Noise std. (generator input)	**0.2**
Normalization	min–max to **[-1, 1]**
Padding/trim	to fixed length **L**

Table 1. Hyperparameters of the generator, discriminator and Training settings of the mode

Graphic User Interface

We implemented a responsive frontend using semantic HTML5, Bootstrap 5, and modular CSS, prioritizing readability, color contrast, and keyboard navigation. The Database page implements a client-side lookup: a normalized search input maps planet identifiers with the example of a local catalog of Exoplanet Kepler-227 b, with the purpose of the model implementation of Exoplanet Hunters in the future. The website contains our interests and goals in this challenge and information about the database used with estimation of exoplanets temperature and size.

Results and discussion

To evaluate the effectiveness of our model in classifying Kepler light curves into three classes: confirmed exoplanets, candidates and false positives, we trained the model using a balanced and curated dataset consisting of 400 light curves, with a class distribution of 35% confirmed, 30% candidates, and 35% false positives. The training was performed over 200 epochs with a batch size of 16, ensuring the generator could learn to synthesize realistic transit-like signals, while the discriminator learned to distinguish between real and synthetic examples as well as to classify real data into the correct exoplanetary category.

After training, the model was evaluated on a test set of 105 samples. The classification report (Table 2) and confusion matrix (Figure 3) revealed the strengths and current limitations of the approach.

	precision	recall	f1-score	support
0	0.3448	0.2778	0.3077	36
1	0.3137	0.5161	0.3902	31
2	0.4000	0.2632	0.3175	38
accuracy			0.3429	105
macro avg	0.3529	0.3524	0.3385	105
weighted avg	0.3556	0.3429	0.3356	105

Table 2. Classification metrics for the three classifications: 0) False positive, 1) Candidate and 2) Confirmed exoplanet.

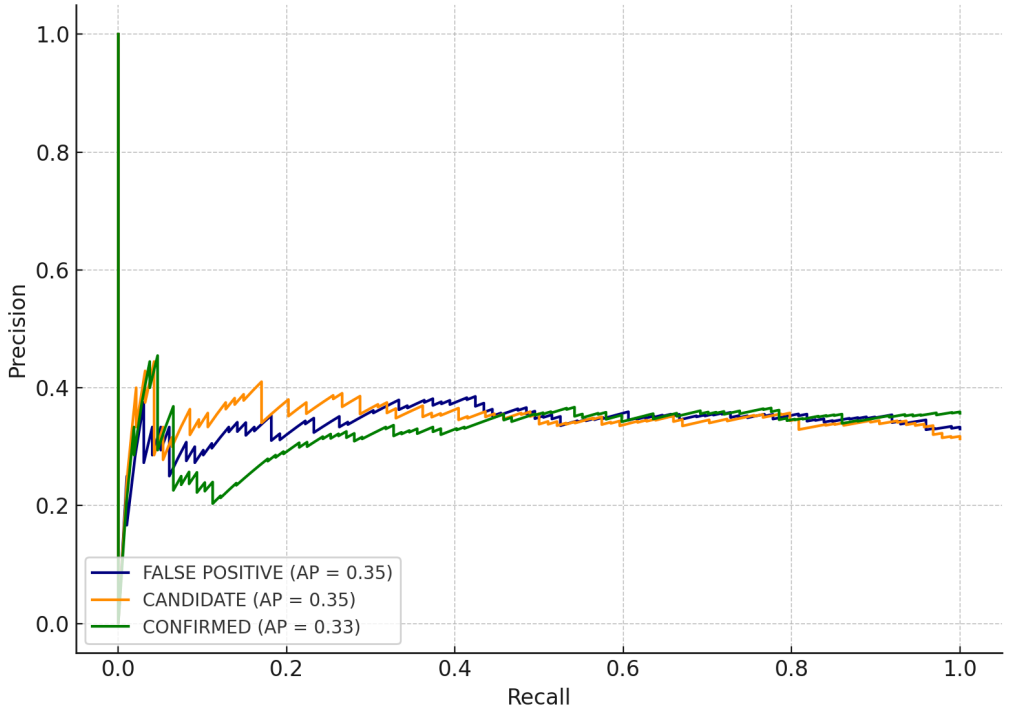


Figure 3. Precision-Recall Curve for Exoplanet Classification

While the overall accuracy (~34%) is relatively low, the model demonstrates that it is learning meaningful representations, particularly given the complexity of the transit signals and the presence of noise or partial curves. The recall was highest for the candidate class (0.516), which suggests that the model was most confident in identifying light curves with uncertain or intermediate characteristics.

The relatively higher precision for confirmed exoplanets (0.400) indicates that the model is slightly better at avoiding false positives when it does predict a planet as confirmed, which is

crucial for exoplanetary science, where false identifications can be costly. However, the low recall for confirmed exoplanets (0.263) shows that many real exoplanets are being misclassified—most often as candidates.

A possible explanation for this confusion lies in the nature of light curves themselves: many candidate and false positive light curves mimic the features of confirmed transits in noisy or incomplete forms. This makes the classification task inherently difficult, especially for a model trained in a semi-supervised adversarial framework.

Moreover, we obtained a confusion matrix (Figure 4) that shows that the model correctly classified 10 confirmed exoplanets, 6 candidates, and 10 false positives. A significant proportion of samples across all classes were misclassified as candidates. Specifically, 17 false positives, 16 candidates, and 18 confirmed exoplanets were predicted as candidates.

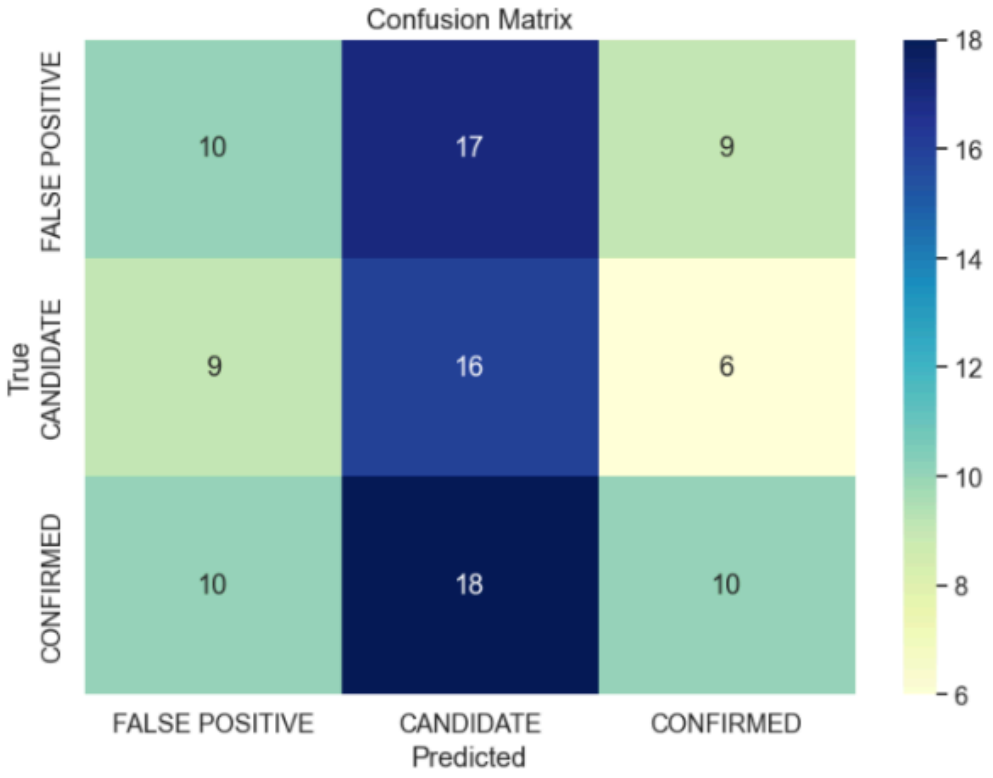


Figure 4. Confusion matrix for our trained model.

This tendency suggests that the candidate class became the model's default classification for ambiguous or borderline signals, possibly due to overlapping features in the transit depth and noise characteristics between candidates and other classes.

Importantly, the generator in EXOSGAN played a critical role in producing synthetic light curves that enriched the training set with diverse examples. This forced the discriminator to develop a more robust feature space, differentiating not only between real and fake light curves but also between the subtle differences across the three real-world classes.

Conclusions

Our project tackles exoplanet discovery as a big-data problem by replacing slow, manual light-curve vetting with an ML pipeline centered on a GAN-assisted 1-D CNN classifier. Using NASA KOI data and Lightkurve, the model was trained on 400 light curves, which matched our results. The class split was 35% confirmed, 30% candidate, 35% false positive, and the model was evaluated on 105 test samples. It achieved ~34% overall accuracy, with its strongest signal on the candidate class (recall = 0.516). For confirmed planets, it was conservative (precision = 0.400) but missed many true cases (recall = 0.263). The confusion matrix shows correct identifications of 10 confirmed, 6 candidate, and 10 false positives, with many ambiguous series pushed toward "candidate" (misclassified as candidate: 17 FP, 16 candidate, 18 confirmed). These outcomes indicate the model is learning physically meaningful transit features while revealing where ambiguity and noise still dominate decisions.

This matters, and it's why we're excited to contribute, because this small quest that produces transparent gains in automated classification can scale to millions of light curves, with reproducible evidence that moves us closer to finding new worlds and doing it as a team with open tools makes the search faster, fairer, and more impactful in how we approach predictions for new space discoveries.

Data Availability

All data and implementation of the model are available in the link:
https://github.com/Saperz4002/NanitosNASA2025

References

1. Skye, A. (2024). Exoplanet Detection and Characterization with Latest Techniques. Retrieved from https://beyondtmrw.org/space-exploration/exoplanetary-research/exoplanet-detection-and-characterization-with-latest-techniques/

2. Carney, S. & Logreira, D. (2025). *Exoplanets*. Retrieved from https://science.nasa.gov/exoplanets/
3. Agnes, C., Naveed, A. & O Chacko, A. (2021). ExoSGAN and ExoACGAN: Exoplanet Detection using Adversarial Training Algorithms. Retrieved from https://arxiv.org/pdf/2207.09665
4. NASA. (s. f.). A *World Away Hunting for Exoplanets with AI*. Retrieved from https://www.spaceappschallenge.org/2025/challenges/a-world-away-hunting-for-exoplanets-with-ai/
5. Karimi, R., Mousavi-Sadr, M., Zhoolideh, M. & Tabatabaei, F. (2025). *Machine Learning for Exoplanet Detection: A Comparative Analysis Using Kepler Data*. Retrieved from https://doi.org/10.48550/arXiv.2508.09689
6. NASA Exoplanet Archive. (n.d.). *Cumulative Exoplanet Table* [Dataset]. IPAC / Caltech. Retrieved October 5, 2025, from https://exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbls&config=cumulative
7. Lightkurve Collaboration, Cardoso, J. V. d. M., Hedges, C., Gully-Santiago, M., Saunders, N., Cody, A. M., Barclay, T., Hall, O., Sagear, S., Turtelboom, E., Zhang, J., Tzanidakis, A., Mighell, K., Coughlin, J., Bell, K., Berta-Thompson, Z., Williams, P., Dotson, J., & Barentsen, G. (2018). *Lightkurve: Kepler and TESS time series analysis in Python* [Computer software]. Astrophysics Source Code Library. Retrieved from http://ascl.net/1812.013
8. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Zheng, X. (2016). *TensorFlow: Large-scale machine learning on heterogeneous systems*. Software available from https://www.tensorflow.org/
9. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). *Scikit-learn: Machine learning in Python.* Journal of Machine Learning Research, 12, 2825–2830. https://jmlr.org/papers/v12/pedregosa11a.html
10. OpenAI. (2025). *ChatGPT (GPT-5)* [Large language model]. https://chat.openai.com