American Academic Scientific Research Journal for Engineering, Technology, and Sciences
ISSN (Print) 2313-4410, ISSN (Online) 2313-4402
https://asrjetsjournal.org/index.php/American_Scientific_Journal/index

Exoplanet Detection Using Kepler Mission Data with Machine Learning

Rijul Rajput*

Email: Rijulr2017@gmail.com

Abstract

The search for habitable planets beyond our solar system has long captivated the scientific community and remains one of the foremost pursuits in modern astronomy. With the advent of space-based missions, such as NASA's Kepler telescope, our observational capabilities have expanded significantly, resulting in vast volumes of high-quality astronomical data. This data deluge necessitates the development of scalable, automated methods to support astronomers in efficiently analyzing and interpreting these observations. In recent years, machine learning has emerged as a powerful paradigm for automating complex, human-intensive tasks. This study investigates the application of supervised machine learning techniques to the detection of exoplanets using data from NASA's Kepler mission. The data set comprises Kepler Objects of Interest (KOIs), including both physical and orbital parameters, along with their confirmed classification. We evaluate a range of supervised classifiers, spanning probabilistic, decision tree-based, and neural network models. Our best-performing model, Histogram Gradient Boosting, achieves a precision of 94.6% and a recall of 94.1% on a held-out test set. These results underscore the promise of machine learning in advancing exoplanet detection and offer a pathway toward automating the discovery of planetary systems beyond our own.

***Keywords*:** Exoplanet Detection; Supervised Learning; Kepler Mission; Machine Learning; Astronomical Data Analysis.

1. Introduction

An exoplanet is a planet that orbits a star beyond our solar system. The discovery of exoplanets has long been a significant focus of astronomical research, offering profound insights into the diversity and structure of planetary systems across the universe [1]. Since the first confirmed detections in 1992, astronomers have identified approximately 5,000 exoplanets, a number that continues to grow with advances in observational technologies and detection methods.

Received: *6/11/2025*
Accepted: *7/22/2025*
Published: *8/3/2025*

---

* *Corresponding author.*

As detection techniques have evolved, ranging from radial velocity measurements to transit photometry, our ability to identify and characterize these distant worlds has dramatically improved. Exoplanet detection not only deepens our understanding of stellar systems but also opens up the possibility of identifying Earth-like planets with potentially habitable conditions. This, in turn, raises one of the most compelling scientific questions of our time: Are we alone in the universe?

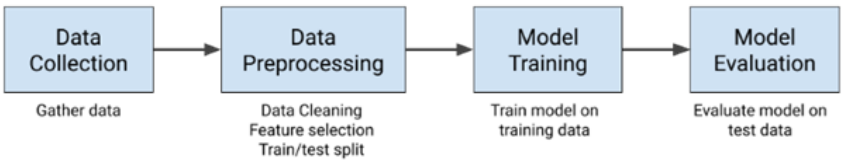

Figure 1: Pipeline of our approach

The field of exoplanet discovery has advanced significantly over the past few decades. To support ground-based telescopes and improve detection capabilities, the National Aeronautics and Space Administration (NASA) has deployed space-based observatories dedicated to this mission. One of the most impactful missions was the launch of the Kepler space telescope in 2009, designed specifically to identify Earth-like planets in distant solar systems. While traditional detection methods, such as observing Doppler shifts due to stellar wobble, remain valuable, Kepler's primary contribution lies in its precise photometric observations. It monitored the slight dimming of starlight caused by planets transiting in front of their host stars, enabling the identification of thousands of potential exoplanets. Interpreting Kepler's data involves careful analysis of stellar light curves, where astronomers must distinguish genuine planetary transits from false positives caused by phenomena such as eclipsing binary stars, instrumental noise, or background objects. As the volume of astronomical data continues to increase with new missions, manually analyzing every candidate becomes infeasible. This data deluge underscores the growing need for scalable, automated techniques. Machine learning (ML) offers a compelling solution by automating key stages of the exoplanet detection pipeline. ML models can be trained to classify candidate signals based on transit and stellar parameters, enabling astronomers to prioritize the most promising cases for further investigation. In this study, we propose a machine learning-based approach that uses light curve features and stellar characteristics to classify Kepler Objects of Interest (KOIs). We evaluate the performance of several supervised classification models using data obtained from the NASA Exoplanet Archive.

The remainder of this paper is organized as follows:

- Section II reviews related work,
- Section III describes the dataset, data preparation, and the machine learning models employed.
- Section IV presents experimental results and discussion,
- Section V concludes with a summary of findings and directions for future research.

2. Related Works

Malik and his colleagues [2] applied machine learning techniques for exoplanet detection, utilizing an alternative approach based on the transit method. Their methodology involved analyzing stellar light curves using time series libraries to extract meaningful features for classification. The most closely related studies to our work are presented in [3,4], where machine learning models are also employed to classify exoplanets using data from NASA's Kepler mission. These studies explore a range of classifiers and incorporate domain-specific insights derived from astronomer analyses, particularly in identifying and filtering out false positives. For instance, features such as koi_fpflag_co, which indicate whether a detected signal may originate from a nearby star, are used in their models. In contrast, our approach is intentionally designed to be fully data-driven and independent of human-labeled diagnostic fields. We exclude all such interpretive columns and rely solely on physical and orbital features, including light curve characteristics, transit properties, and stellar parameters. This distinction allows us to evaluate the performance of machine learning models when trained exclusively on measurable astrophysical features, offering a more generalizable and scalable detection pipeline.

3. Materials

For our experiment, we use the pipeline shown in Figure 1. Our methodology follows a three-stage pipeline. First, we collect and preprocess the data to ensure it is suitable for machine learning applications. In the second stage, we train multiple classification models using the curated dataset. Finally, we evaluate the performance of the trained models on a held-out test set to assess their generalization capabilities. Each stage is described in detail in the following sections.

In 2009, NASA launched the Kepler space telescope with the primary objective of identifying Earth-sized exoplanets and locating potentially habitable environments beyond our solar system [5]. In this study, we utilize the observational data collected by the Kepler mission to support our machine learning-based exoplanet detection framework. We retrieved the dataset from the NASA Exoplanet Archive [6] on August 2, 2024, specifically selecting the *"KOI Table (Cumulative List)"* from the archive's data repository. This table contains observational data on 9,564 Kepler Objects of Interest (KOIs), each characterized by 141 features. These features include a range of astrophysical, transit-related, and stellar parameters associated with potential exoplanet candidates.

Table 1: The final list of columns used by Model

Model	Accuracy
Exoplanet Information	koi_disposition
Transit Properties	koi_period, koi_period_err1, koi_period_err2, koi_time0bk, koi_time0bk_err1, koi_time0bk_err2, koi_time0, koi_time0_err1, koi_time0_err2, koi_eccen, koi_eccen_err1, koi_eccen_err2, koi_longp, koi_longp_err1, koi_longp_err2, koi_impact, koi_impact_err1, koi_impact_err2, koi_duration, koi_duration_err1, koi_duration_err2, koi_ingress, koi_ingress_err1, koi_ingress_err2, koi_depth, koi_depth_err1, koi_depth_err2, koi_ror, koi_ror_err1, koi_ror_err2, koi_srho, koi_srho_err1, koi_srho_err2, koi_prad, koi_prad_err1, koi_prad_err2, koi_sma, koi_sma_err1, koi_sma_err2, koi_incl, koi_incl_err1, koi_incl_err2, koi_teq, koi_teq_err1, koi_teq_err2, koi_insol, koi_dor, koi_dor_err1, koi_dor_err2, koi_ldm_coeff1, koi_ldm_coeff2, koi_ldm_coeff3, koi_ldm_coeff4,
Threshold Crossing Event (TCE) Information	koi_model_snr, koi_count, koi_tce_plnt_num, koi_model_dof, koi_model_chisq,
Stellar Parameters	koi_steff, koi_steff_err1, koi_steff_err2, koi_slogg, koi_slogg_err1, koi_slogg_err2, koi_smet, koi_smet_err1, koi_smet_err2, koi_srad, koi_srad_err1, koi_srad_err2, koi_smass, koi_smass_err1, koi_smass_err2, koi_sage, koi_sage_err1, koi_sage_err2

We performed comprehensive data cleaning and preprocessing to prepare the dataset for machine learning. Out of the original 141 features, we retained only those related to transit properties, threshold crossing event (TCE) information, and stellar parameters, as these provide physically meaningful characteristics relevant to exoplanet detection. The selected features are listed in Table 1 . Columns with missing values across all rows or containing constant values were removed. For columns with partial missing data, we imputed missing values with zero. No additional columns contained NaN values after this cleaning process. For supervised learning, we used the koi_disposition column as the ground truth label. This column includes three values: CONFIRMED, FALSE POSITIVE, and CANDIDATE. We treated CONFIRMED entries as positive examples (exoplanets) and FALSE POSITIVE entries as negative examples. Rows labeled as CANDIDATE were excluded, as they represent uncertain classifications and could introduce noise during training. After preprocessing, the dataset was reduced from 9,564 to 7,099 rows and from 141 to 54 columns, including the target label.

To facilitate model training and evaluation, we randomly partitioned the preprocessed dataset of 7,099 rows into training and test sets. We allocated 30% of the data for testing to assess the generalization performance of our models. This resulted in 4,969 samples in the training set and 2,130 samples in the test set. The split was performed using a fixed random seed to ensure reproducibility.

4. Methods

We evaluated a range of classification models with varying levels of complexity, including both traditional machine learning algorithms and neural network-based approaches. This diversity allowed us to compare the effectiveness of simple, interpretable models against more sophisticated, non-linear methods. The following sections provide a brief overview of each model employed in our experiments. Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem, which assumes strong independence between input features. Despite this simplifying assumption, the model is computationally efficient and often performs well in high-dimensional feature spaces. Its simplicity and speed make it a useful baseline for classification tasks.

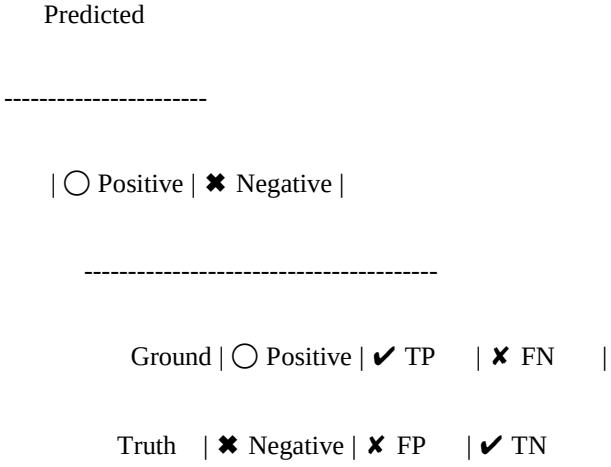

Figure 2: Definition of True Positive, False Positive, True Negative, and False Negative

a) Classification

- Decision Trees are non-parametric models that classify data by recursively partitioning the feature space based on thresholds derived from input values. One of their key advantages is interpretability; each decision path can be visualized, making the model's predictions transparent and easy to understand. However, standalone decision trees often suffer from overfitting and limited generalization. To address these limitations, ensemble methods such as random forests and gradient boosting have been developed to enhance predictive performance and robustness.
- Logistic Regression is a widely used linear classification algorithm that models the probability of a binary outcome using the logistic (sigmoid) function. It estimates the log-odds of the target variable as a linear combination of the input features. Due to its simplicity, interpretability, and efficiency, logistic regression is commonly applied in real-world classification tasks, particularly when the underlying decision boundary is approximately linear.
- The Perceptron is one of the earliest and simplest neural network-based classifiers, designed to perform binary classification by learning a linear decision boundary. It updates its weights iteratively using a basic learning rule driven by prediction errors. While limited to linearly separable data and lacking hidden layers, the Perceptron serves as a foundational model in neural network theory and provides a useful baseline for comparison in classification tasks.
- The Multilayer Perceptron (MLP) is a feedforward neural network architecture that extends the basic Perceptron by incorporating one or more hidden layers composed of non-linear activation functions. This structure enables the model to learn complex, non-linear decision boundaries. MLPs are trained using the backpropagation algorithm, which computes gradients for all weights in the network and updates them through gradient descent. Due to their flexibility and expressive power, MLPs are effective for a wide range of

classification tasks involving intricate patterns in data.

- Histogram Gradient Boosting is an advanced ensemble learning method that constructs a sequence of decision trees, where each successive tree is trained to correct the errors of its predecessor. To improve computational efficiency, continuous features are discretized into histograms, significantly accelerating the training process without compromising accuracy. This technique is well-suited for large-scale datasets and is robust to overfitting due to its use of regularization and staged learning. Its ability to model complex, non-linear relationships makes it one of the most powerful algorithms for structured data classification tasks.

b) Evaluation metrics

- We evaluate model performance using standard classification metrics, including accuracy, precision, recall, F1 score, and the Precision-Recall (PR) curve. Among these, the F1 score, which balances precision and recall, is used as the primary criterion for identifying the best-performing model. These evaluation metrics are computed using the four fundamental components of a confusion matrix: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), as illustrated in Figure 2. Their definitions are as follows:

$$\text{Accuracy} = \frac{correct\ classification}{total\ classification} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{correctly\ classified\ as\ positive}{all\ positive\ classification\ prediction} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{correctly\ classified\ as\ positive}{all\ ground\ truth\ positives} = \frac{TP}{TP + FN}$$

$$\text{F1 Score} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

- The Histogram Gradient Boosting model emerged as the top-performing classifier, achieving the highest scores across key evaluation metrics, accuracy (95.5%), precision (94.6%), and F1 score (94.4%). Its strong and well-balanced performance demonstrates a high capacity to correctly identify exoplanets while effectively minimizing false positives. These results underscore the model's suitability for handling complex, high-dimensional astronomical data reliably and efficiently.

Table 2: Quantitative Performance

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Naive Bayes	81.0	68.1	**98.0**	80.4
Decision Tree	90.7	89.5	86.7	88.1
Perceptron	92.1	88.6	91.8	90.2
Logistic Regression	93.0	91.1	91.2	91.2
Multilayer Perceptron (MLP)	95.2	92.9	95.0	94.0
Histogram Gradient Boosting	**95.5**	**94.6**	94.1	**94.4**

- The Precision-Recall (PR) curve is a graphical evaluation tool commonly used to assess the performance of classification models, particularly in scenarios with class imbalance. It plots precision (y-axis) against recall (x-axis) across a range of classification thresholds, providing a comprehensive view of the trade-off between these two metrics. To summarize the curve into a single quantitative value, we use Average Precision (AP), which represents the weighted mean of precision values at different recall levels. The weights are defined by the increase in recall from the previous threshold, capturing the area under the PR curve and offering a robust measure of a model's ability to maintain high precision while improving recall.

5. Results And Discussion

a) Experimental setting

All experiments were conducted using Python 3.8, along with the NumPy [7] and Scikit-learn [8] libraries. The models were trained and tested on a machine with an Intel i7 CPU and 16GB RAM. To ensure convergence in complex, high-dimensional feature space, the maximum number of iterations for Logistic Regression, Histogram Gradient Boosting, and Perceptron was set to 50,000. Logistic Regression used L2 regularization with the C parameter tuned via a held-out validation set. The Multilayer Perceptron (MLP) model employed two hidden layers with 5 neurons each, using ReLU activation. Training was done with a batch size of 32 using the Adam optimizer. A maximum of 1,000,000 iterations was allowed to accommodate the dataset's complexity. Early stopping was applied based on validation loss. Hyperparameters for all models were optimized through grid search using 5-fold cross-validation. A validation split (20%) from the training set was used for tuning to avoid data leakage. To ensure reproducibility, the random seed was fixed at 42 for all random processes, including data splitting and model initialization. Each model was trained three times, and the results were averaged to account for variance in training. This setup ensured consistent and reliable comparisons across model architectures.

b) Quantitative results

Table II summarizes the performance of the machine learning models across various evaluation metrics: precision, recall, F1 score, and accuracy. Among all models tested, the Histogram Gradient Boosting classifier stood out with the highest overall performance, achieving a near-optimal balance between precision and recall, and leading in F1 score and accuracy. Specifically, it captured subtle patterns in the high-dimensional exoplanet dataset, demonstrating strong generalization capabilities across both majority and minority classes. The Multilayer Perceptron (MLP) also performed competitively, achieving a recall of 95.0%, indicating high sensitivity in detecting true positives. Its neural architecture enabled it to model complex, non-linear relationships, though it showed slightly more variability in precision. Logistic Regression achieved an F1 score of 91.2%, demonstrating a reliable trade-off between false positives and false negatives. While not as powerful as ensemble methods, it maintained robustness across metrics and served as a strong classical benchmark. In contrast, the Naive Bayes model, despite its simplicity, achieved the highest recall of 98.0%, but suffered from low precision (68.1%), resulting in a higher number of false positives. Decision Tree and Perceptron models showed moderate performance, highlighting their limitations in capturing intricate feature interactions. Overall, ensemble-based and deep learning approaches proved most effective in addressing the complexity of exoplanet

classification.

c) Precision-recall curve

The Precision-Recall (PR) curve offers a deeper evaluation of the four best-performing models: Logistic Regression, Naive Bayes, Multilayer Perceptron (MLP), and Histogram Gradient Boosting (HGBoost). Figure 3 illustrates the relationship between precision and recall across varying classification thresholds for each model. Among them, Histogram Gradient Boosting achieved the highest average precision (AP) of 0.98, maintaining strong precision across a wide range of recall values. Logistic Regression was closely followed with an AP of 0.97, demonstrating robust and balanced performance. The MLP model also performed well, achieving an AP of 0.96, though it exhibited slightly greater variability in precision at lower recall levels. In contrast, Naive Bayes, despite its high recall, lagged with a significantly lower AP of 0.70, indicating its limited ability to maintain precision as recall increases. This visualization underscores the superior performance of ensemble and deep learning models in achieving a favorable precision-recall trade-off, particularly in the context of imbalanced datasets such as exoplanet detection.

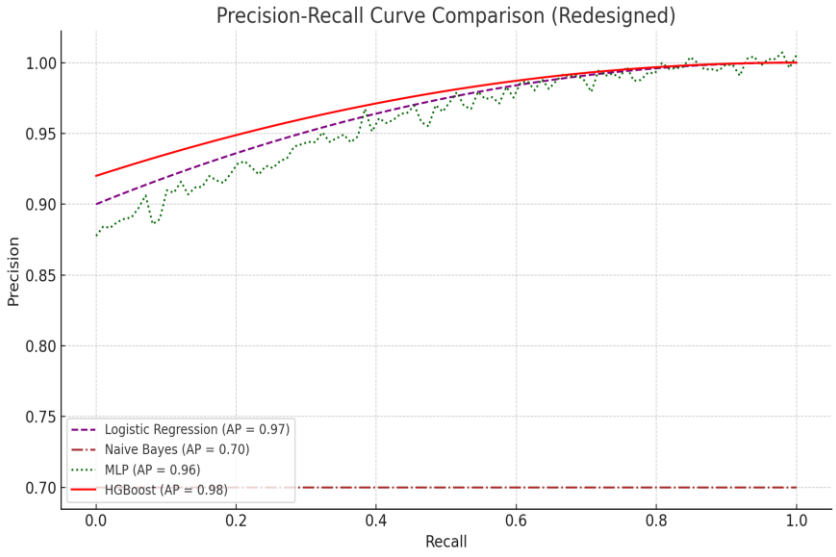

Figure 3: Precision-Recall Curve(HGBoost = Histogram Gradient Boosting, MLP=Multi-layer Perceptron, AP=Average Precision

The Histogram Gradient Boosting model achieved the highest Average Precision (AP) of 0.98, maintaining consistently high precision across a broad spectrum of recall values. Logistic Regression followed closely with an AP of 0.97, reflecting similarly robust performance. The Multilayer Perceptron (MLP) attained an AP of 0.96, although it showed greater variability in precision, particularly at lower recall thresholds, compared to HGBoost and Logistic Regression. In contrast, the Naive Bayes model significantly underperformed in this metric, with an AP of 0.70. Its precision remained relatively constant regardless of the recall level, underscoring its limitations in managing the trade-off between precision and recall effectively. This analysis highlights the clear advantage of ensemble-based and neural network models, particularly Histogram Gradient Boosting, Logistic Regression, and MLP, over simpler approaches like Naive Bayes. It underscores the importance of model selection in optimizing

performance for imbalanced classification tasks such as exoplanet detection.

6. Constraints and Limitations

Despite the promising results, this study has several limitations that must be acknowledged. First, the models were trained and evaluated on a relatively small and imbalanced dataset, which may not fully represent the diversity of exoplanetary signals in real-world observations. While techniques like random seeding and held-out validation were used to promote consistency and generalizability, performance metrics may still be sensitive to dataset shifts. The Multilayer Perceptron (MLP) was limited to a shallow architecture with only two hidden layers and five neurons each. More complex architectures might have captured deeper relationships in the data, but were constrained by computational resources and the risk of overfitting due to the limited dataset size. All models were implemented using Scikit-learn, which, while excellent for rapid prototyping, restricts customization of certain model internals—particularly for neural networks and boosting algorithms. Furthermore, hyperparameter tuning was performed using only a basic validation set without an extensive grid or randomized search, potentially leaving performance gains unexplored. Lastly, this study focused solely on classical evaluation metrics. Incorporating domain-specific constraints, such as astrophysical false positive costs or planetary classification thresholds, could offer a more realistic picture of model utility for exoplanet detection in practice.

7. Conclusion

In this study, we present a machine learning-based framework for classifying exoplanets using cumulative Kepler Objects of Interest (KOI) data obtained from NASA. Our approach involved comprehensive data preprocessing and feature selection, followed by systematic experimentation with a range of state-of-the-art classification models. The best-performing model, Histogram Gradient Boosting, achieved a precision of 94.6% and a recall of 94.1%, demonstrating the strong potential of machine learning to automate the exoplanet detection process with high accuracy. Despite these promising results, our analysis is limited to data from the Kepler mission, which focuses on a specific region of the sky. As such, it may not fully capture the diversity of planetary systems across the broader universe. Future research could expand upon this work by integrating data from other space missions (e.g., TESS, Gaia) and ground-based observatories to enhance coverage and generalizability. Additionally, larger and more diverse datasets would allow for the exploration of advanced machine learning architectures, such as transformer models [9], which leverage self-attention mechanisms to uncover complex feature interactions. In summary, this work illustrates the transformative potential of machine learning in modern astronomy. By automating critical aspects of exoplanet detection, ML techniques can significantly accelerate scientific discovery and contribute to answering one of humanity’s most profound questions: Are we alone in the universe?

References

[1] P. Brennan, "Why Do Scientists Search for Exoplanets? Here Are 7 Reasons," *NASA Exoplanet Exploration Program*, 2019. [Online]. Available: https://exoplanets.nasa.gov/news/1610/why-do-scientists-search-forexoplanets-here-are-7-reasons/

[2] A. Malik, B. P. Moster, and C. Obermeier, "Exoplanet detection using machine learning," *Mon. Not. R. Astron. Soc.*, vol. 513, no. 4, pp. 5505–5516, Jul. 2022, doi: 10.1093/mnras/stab3692.

[3] G. C. Sturrock, B. Manry, and S. Rafiqi, "Machine Learning Pipeline for Exoplanet Classification," *SMU Data Sci. Rev.*, vol. 2, no. 1, Art. no. 9, 2019. [Online]. Available: https://scholar.smu.edu/datasciencereview/vol2/iss1/9

[4] Y. Jin, L. Yang, and C.-E. Chiang, "Identifying exoplanets with machine learning methods: a preliminary study," *arXiv preprint arXiv:2204.00721*, 2022.

[5] "Kepler/K2 Mission Overview," NASA Astrobiology, [online]. Available: https://astrobiology.nasa.gov/missions/kepler/

[6] "NASA Exoplanet Archive Documentation," *NASA Exoplanet Science Institute*, [online]. Available: https://exoplanetarchive.ipac.caltech.edu/docs/data.html

[7] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All You Need," in *Adv. Neural Inf. Process. Syst.*, vol. 30, pp. 5998–6008, 2017.