scientific reports

Check for updates

OPEN

Training a convolutional neural network for exoplanet classification with transit photometry data

Juliana Wang

The search for exoplanets aims to identify planets with compositions similar to Earth's, providing insights into planetary formation and habitability. As a result, efforts to enhance the efficiency of exoplanet research have led to the development of various detection methods, including transit photometry. Despite their effectiveness, these methods produce data that require detailed interpretation, such as identifying dips in light curves. Machine learning has then emerged as a powerful alternative, offering rapid image classification and the ability to analyze complex datasets in a short span of time. This paper applies a convolutional neural network (CNN) to the Kepler dataset, which consists of time-series light curve data from the Kepler Space Telescope, used for detecting exoplanets through transit events. The final CNN architecture, with hyperparameters set as (300, 200, 200, 100, 100), was identified as the best-performing model after evaluating multiple configurations. These results highlight the model's strengths and areas for improvement; while it excels at identifying false positives (low miss rate of 5%), its higher miss rate for the 'CONFIRMED' class (40%) suggests a need for better detection of true exoplanets. The AUC score of 0.91 further underscores the model's strong overall performance.

Keywords Exoplanet detection, Neural networks, Computational astrophysics, Machine learning

In hopes of finding new habitable zones, new forms of life, and to better understand the origins of the universe, scientists prioritize exoplanet research[1]. The first confirmed near-Earth-size exoplanet orbiting within the habitable zone of a Sun-like star is Kepler-452b[2]. Since the 1990s, researchers have detected thousands of exoplanets using methods such as radial velocity (measuring Doppler shifts in a star's spectral lines), transit photometry (observing dips in a star's brightness), direct imaging (blocking starlight to capture planet images), gravitational microlensing (detecting light bending from distant stars), and astrometry (tracking star movements)[3]. According to the NASA Exoplanet Archive[4], as of January 23, 2025, the numbers of detected exoplanets are 1,096 with radial velocity, 4,329 with the transit method, 50 with direct imaging, 232 with microlensing, and 3 with astrometry. Notably, the first exoplanet detected around a Sun-like star, 51 Pegasi b, was discovered using the radial velocity technique[5]. However, to process data and draw conclusions more efficiently, machine learning (ML) algorithms have recently been employed to classify images and visual patterns from observatories, aiding in the identification of planetary motion. For instance, researchers at Princeton University developed an artificial intelligence model that predicts the stability of planetary systems by analyzing orbital configurations, significantly outperforming previous methods: "Our approach significantly outperforms previous methods based on systems' angular momentum deficit, chaos indicators, and parameterized fits to numerical integrations"[6]. Similarly, a recent study utilized ML techniques to map stable orbital regions around hypothetical planets, further showcasing the potential of these algorithms in celestial mechanics[7].

As a way to organize data and conduct more thorough analysis to obtain results, data classification has been implemented to achieve this efficacy. With the invention of ML taking place, this set of algorithms have allowed models to learn through data sets without needing direct instruction, improving its results through training for a specific amount of time. This specific application of artificial intelligence (AI) and deep learning has now been continuously developed and used throughout studies up until this day, to classify extraterrestrial and interstellar data in many papers[6,8,9]. Combining ML and data classification to sort raw exoplanet observation data enhances the clarity of patterns and characteristics, enabling more efficient model training and observation of processes.

This study utilizes the publicly available Kepler dataset, which contains over 10,000 light curves. Among these, approximately 2700 entries are labeled as confirmed exoplanets or planet candidates, while the remainder are classified as false positives or unclassified objects[10]. The data set can be accessed via the NASA exoplanet archive[11]. A neural network (NN) is employed to classify this data and evaluate its potential for identifying future

Polygence, São Paulo, Brazil. email: jewbmewb@gmail.com

exoplanet candidates. While previous studies have demonstrated that NNs achieve high accuracy in exoplanet classification tasks—such as Jin et al.[8], which reported a peak accuracy of 99.79%—this research focuses on testing different network architectures by varying the number of layers to determine the optimal configuration for performance.

Additionally, this paper provides a detailed description of the methods and processes used in the experiment, focusing on exoplanet detection. The following sections examine the neural network's architecture, configuration, and training procedures, enabling an evaluation of its performance using tailored metrics. The results section reports detailed statistics from multiple trials, highlighting the model's consistency and accuracy. Finally, the study concludes by demonstrating the neural network's effectiveness with expansive data sets (Kepler data set), conveying its potential applications in astrophysics.

Method

Neural networks

For some background on the architecture being implemented, a subset of AI, NNs are computer networks inspired by the structure of the human brain, with each node (denoted as a circle in Fig. 1) representing a neuron, allowing it to process information through connections and passing data through layers/filters[12]. The structure of an artificial NN further allows this specific algorithm to start learning with no prior information, and instead gives it an adaptive structure the more time it is trained. Aside from the input and output layers, NNs contain hidden layers, and especially in the multi-layer perceptron classifier (MLP Classifier), a feedforward NN (a type of artificial NN) comprises an input layer, hidden layers and an output layer, a more basic neural structure. It does, however, use sigmoid neurons to process non-linear data efficiently. Sigmoid neurons are components of neural networks, utilizing the sigmoid activation function to map input values onto a continuous, S-shaped curve between 0 and 1. It introduces non-linearity into the network, enabling it to model complex relationships in data[13]. In the astrophysical context, the ability of NNs to detect faint and complex signals is particularly advantageous: by analyzing raw light curves from missions like Kepler, the models can identify subtle variations caused by planetary transits that may be indistinguishable using traditional statistical methods. Additionally, features like the ReLU (Rectified Linear Unit) activation function, often used in CNN architectures, enable the model to handle sparse data more effectively, making it robust for the noisy and incomplete datasets common in astronomical surveys[14].

There are certain types of artificial NNs: convolutional neural networks (CNN), and recurrent neural networks (RNN). The difference between the two models mainly lies in their ability to process data; CNNs are

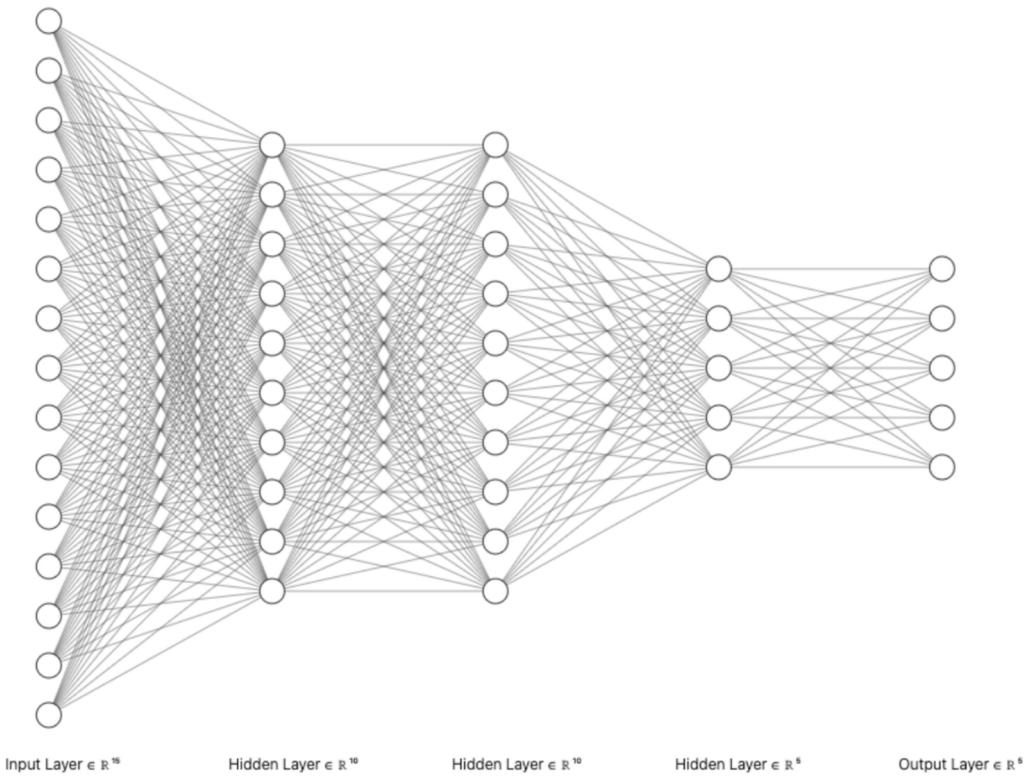

Fig. 1. Visualization of the CNN architecture of 5 dimensions with layers configured as (300, 200, 200, 100, 100), divided by 20. In the diagram, circles represent the nodes within each layer, and lines represent the connections between the layers, illustrating the flow of information through the network. Figure generated through NN-SVG. This architecture was selected based on its superior performance, achieving a mean accuracy of 92.5% on test data.

better suited for data with spatial data like images (and provides an advantage to a NN as it delves deeper into complex features), whereas RNNs are applied for sequential data[15]. This paper utilizes the MLPClassifier, an artificial NN. Different variations of layers have been modified within the model (e.g. 2-dimensional and three dimensional, differing layer numbers) to obtain the most accurate model. The accuracy is then represented through a Receiver Operating Characteristic (ROC) curve.

A ROC curve is a visual plot that shows the performance of a binary classifier system (such as a NN ML algorithm) by showing its accuracy through the area covered under the curve line formed (typically a high recall rate produces better results). It is calculated by inputting the true positive rate against the false positive rate. The ROC curve was selected as the primary evaluation metric for this study due to its effectiveness in assessing the model's ability to differentiate between exoplanet candidates and non-candidates, even in the presence of imbalanced data sets like Kepler's. The Area Under the Curve (AUC) quantifies the overall performance of the model by measuring the total area under the ROC curve. A higher AUC value (closer to 1) indicates better model performance, as it reflects a higher true positive rate and a lower false positive rate.

Data definitions

Having established the functioning process of this model, the data definitions used for this study are described below.

For the architecture of our NN, hyperparameters (the layer and dimension) of our NN are modified by trial to identify the optimum number of layers for best accuracy, and the model type that has the highest area coverage under the ROC curve, and satisfactory recall and precision rates (≥ 0.6) are analyzed. The Kepler data set contains 50 columns and 9564 rows. However, only 'CONFIRMED' and 'FALSE POSITIVE' data points are used to test out the model, meaning that rows with 'CANDIDATE' status are eliminated, which fall under the koi_disposition row. (koi_disposition—the literature of an exoplanet, can be 'CANDIDATE', 'FALSE POSITIVE', 'NOT DISPOSITIONED' or 'CONFIRMED'[11]). To check for definitions of terms in the Kepler Data set, refer to Table 1 below:

Furthermore, empty rows are deleted and 55,188 data points are fed to the network. The data points fed to the NN were numbers under columns 'koi_disposition', 'koi_period', 'koi_impact', 'koi_duration', 'koi_depth', 'koi_prad', 'koi_teq', 'koi_insol','koi_model_snr','koi_steff', 'koi_slogg','koi_srad' (12 columns), and a total of 4599 rows, leading to a total of 55,188 data points.

These columns were classified as "Important" in the algorithm —(columns that did not seem to be relevant to the discovery process were dropped) as these features were chosen to ensure the model captures the most relevant aspects of exoplanetary characteristics (e.g. features such as koi_duration, which reflects the transit length of a planet). As mentioned initially, these patterns are a representation of planetary behavior (dip/change in light), and the numbers allow the machine to understand what values fall under the categorization of "CONFIRMED" and "FALSE POSITIVE". By tracking these two classes, it will help depict the accuracy of the model due to its ability to differentiate the estimations with the results of the labeled data set. Furthermore, due to the presence of some empty rows, instead of filling them up they are removed with the function df_important.dropna, leaving 4599 rows out of the total 9564, which corresponds to about 48% of the Kepler data set. It still provides a substantial amount of data for analysis, amounting to 55,188 data points. The sample size is considered adequate for the analysis as it encompasses a sufficient number of data points to capture significant patterns and facilitate generalization to unseen data. Furthermore, given the inherent structure of the data set, an increase in the sample size would likely result in only marginal improvements in model performance.

Process

The study's MLP Classifier is built using the scikit-learn Python library (version 1.17 for supervised neural network models[17]). The activation function used in the nodes of the model is the ReLU function (used to introduce nonlinearity in a NN), and the solver used was the Adam solver (computationally efficient optimizer suited for problems large in data)[18]. The model is trained on two arrays: one representing the input features (n_samples × n_features) and the other representing the output labels (n_samples). The input features correspond

Name	Definition (derived from Caltech's exoplanet archive website)
'koi_disposition'	values in the data set that are under CANDIDATE, FALSE POSITIVE, NOT DISPOSITIONED or CONFIRMED
'koi_period' (days)	Time in between planetary transits measured in days
'koi_impact' (Dimensionless)	Visible distance between the star's surface and its exoplanet
'koi_duration' (hours)	The duration of the observed planet's transit in hours
'koi_depth' (ppm)	Dip in stellar light/Dim in stellar lightness. Typically computed from data, in parts per million
'koi_prad' (Earth radii)	Radius of the observed planet measured in Earth radii
'koi_teq' (Kelvin)	Equilibrium temperature of the planet measured in Kelvins
'koi_insol' (Earth fluxes)	Incident stellar flux received by the planet (amount of starlight the planet receives), relative to what Earth receives
'Koi_model_snr' (dimensionless)	Signal-to-noise ratio (SNR) of the model's match to the light curve data for a specific planet candidate
'koi_steff' (Kelvin)	Stellar effective temperature of the star in Kelvins
'koi_slogg'(log_{10} (cms^{-2})	Surface gravity of the star (in base-10)
'koi_srad' (solar radii)	Photospheric radius of the star in solar radii

Table 1. Definitions of terms used in the Kepler Data set[16].

to the characteristics of the data (e.g., koi_period, koi_impact, etc.), while the output labels represent the class (e.g., exoplanet candidate or non-candidate). After training, the model is tested, and results are plotted using a library from scikit-learn (version 3.3 for Model Evaluation on the SciKit website[19]) to generate the ROC curve.

After the output is displayed, the results from the end of the trial are plotted on a table. Each trial is run 3 times; Since the performance across the 3 runs did not exhibit significant variability, running the model 3 times was deemed sufficient to obtain a reliable estimate of its performance. This approach ensures statistical stability and avoids excessive computational overhead, which could lead to unnecessary delays.

In the data collection, the following metrics were used to evaluate the performance of the model:

Precision (measures the accuracy of predictions):

$$\frac{TruePositives}{TruePositives + FalsePositives} \tag{1}$$

And to calculate the recall (accounts for the total amount of predictions):

$$\frac{TruePositives}{TruePositives + FalseNegatives} \tag{2}$$

The layers are modified throughout trials (starting by (50,50)), and follows the visual structure in Fig. 1. In Table 2 (below), summary statistics for all the final set of used features are included below (data collection will be based off of this configuration). Through an observation of the summary statistics, it can be determined how well the model performs and enable model comparison by providing a standardized view of their performance, such as average accuracy. Units in each row are presented relative to the definitions provided in Table 1, and allow for scale interpretation, highlighting important changes such as changes in starlight for 'koi_depth'. Unusual values and outliers will also display data and model issues.

Data collection table

16 trials were conducted to figure out the most accurate layer configuration.

The trial starts with two dimensions. The first three trials had layer numbers that added up to 100, and to avoid overfitting instead of adding a larger layer amount another dimension would be included. Trials 4–7 are then increased by increments of 100, along with trials 8–11 (layer amounts would start from 100 and gradually increase until a fallout). The rest of the layers only have the layer amount adjusted in a limited number of dimensions to avoid overfitting caused by excessive layer and dimension numbers.

Results

Building on these findings, trial 15 exhibits the highest AUC percentage in the ROC curve of 0.91 (see Fig. 2) and achieves precision and recall rates above the 0.6 threshold. These metrics can be better understood when framed in the context of exoplanet research needs; High AUC suggests that the model is effective at distinguishing between 'CONFIRMED' and 'FALSE POSITIVE' cases, which is critical for prioritizing telescope time. For instance, when deciding which candidates to follow up with more resource-intensive observations, a high AUC ensures that the top-ranked candidates are likely to be true positives. However, the precision/recall values ≥ 0.6 highlight that while the model performs reasonably well, there is room for improvement in identifying candidates with weaker or unusual signals. Other models such as those with a configuration of 4 dimensions, despite performing similarly well (AUC percentage of 0.9, refer to Figs. 3 and 4), did not meet requirements for precision and recall metrics, which shows that the model has area for improvement and training. This underlines the importance of not solely relying on ML models but instead using them as part of a broader candidate validation pipeline. By refining the model to further improve recall (e.g., reducing the 40% miss rate for 'CONFIRMED' planets), its practical application in exoplanet research (like TESS and PLATO) could be greatly enhanced.

Although Trials 5, 6, 9, 11, 12, and 14 demonstrate high AUC values in the ROC plot, their precision and recall rates do not consistently meet the 0.6 threshold. Despite Trial 11–15 retaining a similar AUC percentage, on Trial 16 the area coverage starts to fall off, potentially meaning that the ML algorithm might have picked up noise from the input data. Therefore, in these trials CNN configurations with 3 dimensions (featuring higher layer sizes such as (400, 400)), 4 dimensions (with smaller layer sizes such as (100, 100, 100, 100)), and 5 dimensions (with moderate layer sizes such as (300, 200, 200, 200, 100)) generally demonstrate stronger performance. Anything

Index	koi_period	koi_impact	koi_duration	koi_depth	koi_prad	koi_teq	koi_insol	koi_model_snr	koi_steff	koi_slogg	koi_srad
25%	2.24	0.22	2.53	184.60	1.52	559.00	23.35	14.70	5320.00	4.21	0.83
50%	8.51	0.58	3.91	507.45	2.68	928.00	176.51	30.75	5779.50	4.44	1.00
75%	36.18	0.92	6.43	2775.10	25.45	1496.25	1201.17	123.10	6126.00	4.54	1.37
count	7316.00	7016.00	7316.0	7016.00	7016.00	7016.00	7057.00	7016.00	7016.00	7016.00	7016.00
max	1071.23	100.81	138.54	1,541,400.00	200,346.00	14,667.00	10,947,554.55	9054.70	15,896.00	5.28	229.91
mean	58.82	0.79	5.87	30,620.12	129.97	1148.59	8485.67	326.63	5727.71	4.30	1.78
min	0.24	0.0	0.11	0.80	0.08	92.00	0.02	0.00	2661.00	0.05	0.12
std	121.08	3.67	6.97	92,873.98	3519.62	898.33	160,221.87	891.67	825.22	0.44	6.20

Table 2. Summary statistics for all final set of used features.

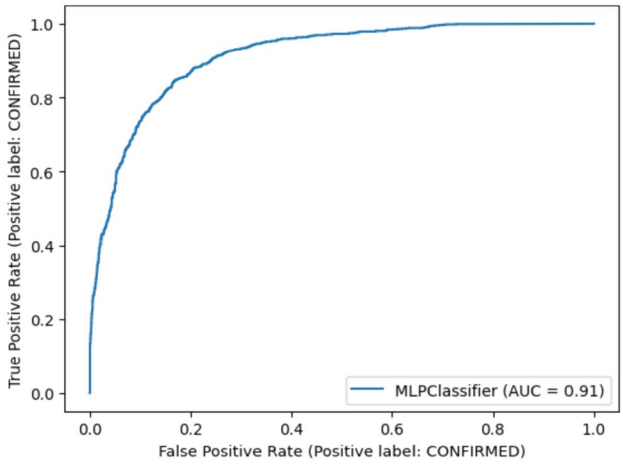

Fig. 2. Roc Curve Output of Trial 15.

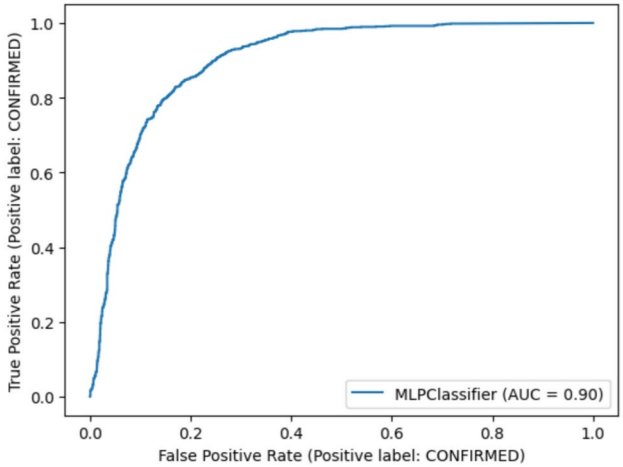

Fig. 3. ROC curve for hyperparameter configuration (300,200,200,100). AUC of 0.9, highlighting its performance under a larger network depth.

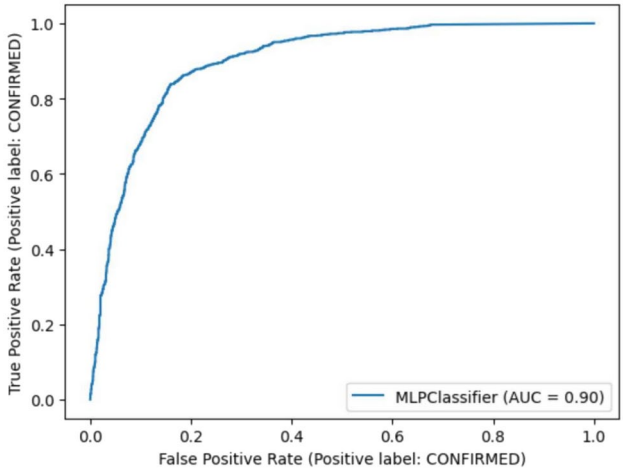

Fig. 4. ROC curve for hyperparameter configuration (100,100,100,100). AUC of 0.9, demonstrates its comparative performance despite lower network complexity.

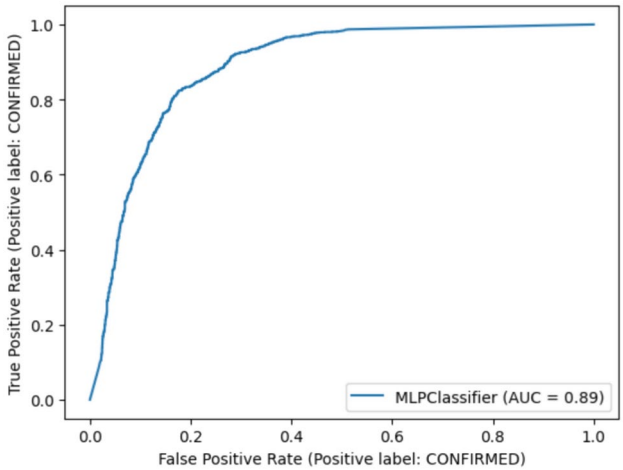

Fig. 5. ROC curve for hyperparameter configuration (500,500). AUC of 0.89, reflects reduced performance with a compact network.

Trial #	Amount of layers	Recall-FALSE POSITIVE	Precision-FALSE POSITIVE	Recall-CONFIRMED	Precision-CONFIRMED	ROC curve area coverage
1	(50,50)	0.65	0.97	0.95	0.57	0.87
2	(70,30)	0.95	0.80	0.50	0.84	0.89
3	(40,60)	0.72	0.95	0.92	0.61	0.88
4	(200,200)	0.68	0.94	0.91	0.58	0.88
5	(300,300)	0.97	0.78	0.45	0.88	0.91
6	(400,400)	0.61	0.97	0.96	0.55	0.90
7	(500,500)	0.77	0.92	0.86	0.64	0.89
8	(100,100,100)	1.00	0.70	0.12	0.98	0.88
9	(200,200,200)	0.98	0.76	0.34	0.91	0.91
10	(300,300,300)	0.94	0.78	0.44	0.77	0.87
11	(400,400,400)	0.81	0.91	0.84	0.69	0.90
12	(100,100,100,100)	0.92	0.83	0.60	0.79	0.90
13	(200,200,200,100)	0.93	0.83	0.62	0.80	0.89
14	(300,200,200,100)	0.67	0.96	0.94	0.58	0.90
15	(300,200,200,100,100)	0.95	0.83	0.60	0.85	0.91
16	(300,200,200,200,200)	0.82	0.88	0.76	0.67	0.87

Table 3. Trial number alongside with trial's result.

slightly different in both dimensions and layers might cause overfitting (the picking up of noise) and anything lower did not seem to perform as high as trials 11–15 in terms of area coverage in ROC Curve (as shown in Fig. 3, despite having more layers than the model in Fig. 4, its AUC percentage did not differ significantly.). Additionally, when compared with simpler configurations (such as those with two dimensions), the AUC percentage slightly falls (drops to 0.89), and precision and recall rates still fall under the set threshold (refer to Fig. 5). The trial 15 model exhibited a 40% miss rate for the "CONFIRMED" class and a 5% miss rate for the "FALSE POSITIVE" class, while misclassifying 15% of "CONFIRMED" instances and 17% of "FALSE POSITIVE" cases. Compared to existing methods, this model's false-positive rate of 17% demonstrates a potential for improvement. However, leveraging the AUC of 0.91, the model can prioritize candidates more accurately, potentially reducing wasted telescope time spent on false positives. To address the limitations, the model could also prioritize candidates by ranking them based on confidence scores derived from the classification probabilities, and employ cross-validation techniques, such as stratified k-fold cross-validation, to assess the model's robustness, mitigate data imbalance effects, and systematically optimize hyperparameters for improved performance.

In Table 3 (below), statistics from the trials of data collection are included—as seen, Trial 15 meets all given criteria compared to the other runs.

Evaluations and limitations

The study's results, while promising with an AUC score of 0.91, revealed limitations that warrant further discussion—leading to the evaluation of some characteristics and limitations the model has. Notably, the model exhibited a 40% miss rate for the 'CONFIRMED' class and misclassification rates of 15% and 17% for the

'CONFIRMED' and 'FALSE POSITIVE' classes, respectively. These issues may stem from data imbalance, where the data set contains more false positives than confirmed planets, which can skew the model's predictions. For reliable automation, particularly in applications where high confidence is essential, precision and recall levels exceeding 90% tend to be necessary. The 40% miss rate for the 'CONFIRMED' class in this study (equivalent to a recall of ~ 60%) falls significantly below this threshold, indicating the need for improvements before automation can be considered viable. Additionally, potential overfitting due to the chosen CNN architecture (300, 200, 200, 100, 100) might hinder its ability to generalize to unseen data. The removal of 52% of rows with missing values could have further introduced bias; if the missing data were not random—such as certain types of exoplanets being more likely to have incomplete information—this could distort the model's learning process and reduce the representativeness of the data set. To address these limitations, future work could incorporate more data sets from missions like TESS or Gaia, employ systematic hyperparameter tuning (e.g., grid search or Bayesian optimization), and explore advanced architectures such as Transformers or ensemble methods to improve classification accuracy.

Moreover, recent studies have demonstrated the growing role of machine learning in exoplanet detection and characterization, providing valuable context for evaluating the limitations of this work. Tamayo et al.[6] introduced the Stability of Planetary Orbital Configurations Klassifier (SPOCK), which trained on 100,000 three-planet systems to classify the long-term stability of compact multi planet systems. In contrast, this study utilized a data set of 58,188 data points from the Kepler mission, where 52% of rows were removed due to missing values, potentially introducing bias and reducing representativeness. Compared to Tamayo et al.[6], the smaller data set and focus on classification rather than stability analysis may have limited the scope of this study. Similarly, Jin et al.[8] achieved high classification accuracies—up to 99.79%—using supervised learning methods such as decision trees and neural networks on the Kepler data set, whereas this study employed a CNN model with an AUC score of 0.91 but faced challenges such as data imbalance and misclassification rates of up to 17%, and missing up to 40% of class 'CONFIRMED' planets. In contrast to the machine learning for cross-correlation spectroscopy (MLCCS) approach introduced by Nath-Ranga et al.[9], which leverages weak assumptions to enhance detection sensitivity for faint exoplanets, this study faced notable limitations in its classification of exoplanets using a CNN. The mentioned study employed perceptrons and one-dimensional CNNs to effectively identify molecular signatures in spectral data, achieving up to 77 times greater detection rates compared to traditional signal-to-noise ratio (S/N) metrics. While this study focuses solely on classification tasks using a CNN architecture, these comparisons highlight how some of the limitations encountered here, such as data set representativeness and model sensitivity, could be addressed.

The present model also did not make use of automated feature importance ranking techniques such as Random Forests; feature selection was performed manually to prioritize interpretability and computational efficiency. This approach demonstrates that even basic machine learning techniques, with carefully chosen features, can contribute to exoplanet classification and makes this method accessible to researchers with limited computational resources. Future work could, however, incorporate feature importance methods to allow for a more systematic identification of which features contribute most to the classification process, potentially revealing interactions or patterns not immediately apparent through manual selection. Jin et al.[8] successfully employed Random Forests to achieve high classification accuracy on exoplanet datasets, and applying this method in future work could refine the feature selection process and improve the overall performance of the model, addressing the low recall rates that were observed during trials.

Lastly, even though prior works[8,9] have demonstrated high classification accuracies using neural networks and alternative machine learning methods, this study provides a unique contribution by focusing on hyperparameter optimization and class-specific performance metrics when applied to raw light curve data from the Kepler mission. By addressing challenges such as data imbalance and misclassification rates, this work highlights practical limitations and opportunities for improving CNN-based exoplanet detection models in real-world scenarios. While the current study did not focus explicitly on edge cases such as planets with unusual orbital properties or weak transit signals, the model's ability to capture nonlinear relationships suggests that it may outperform traditional linear classifiers in these scenarios. Future work could incorporate datasets with labeled edge cases to evaluate this capability. Additionally, retraining the model on more diverse datasets could enhance its sensitivity to weak or atypical signals. Our approach simplifies the detection pipeline by applying CNNs directly to the raw time-series data, demonstrating the potential of deep learning techniques in exoplanet detection without relying on extensive preprocessing steps.

Conclusion

To summarize the findings, in this study, an MLPClassifier, a type of Fully-Connected Neural Network, was employed to classify exoplanets and optimize hyperparameters. Using NASA's Kepler dataset[11] with over 10,000 candidates, the model achieved an AUC score of 0.91 and precision and recall rates of 0.6, when set and tested with optimal hyperparameters of (300, 200, 200, 100, 100) layers. While not fully optimized for autonomous exoplanet classification, the model demonstrates its potential for professional workflows, such as serving as a pre-screening tool for large-scale missions like TESS, where manual inspection of light curves is impractical. This direct application of CNNs to raw light curve data could complement traditional methods like BLENDER or SPOCK. For instance, Tamayo et al.[6] classified the long-term stability of planetary systems, while Nath-Ranga et al.[9] achieved higher detection sensitivity using CNN for spectroscopic data. Building on these approaches, this study highlights the promise and challenges of deep learning in exoplanet detection. Expanding the model to incorporate diverse datasets from missions like TESS or Gaia could improve its generalizability, paving the way for scalable, hybrid human-AI workflows to advance exoplanet discovery.

Looking forward, the applicability of this method could be applied to other data sets, such as those from the TESS space telescope[20]. Given the similarities between Kepler and TESS—both missions aim to detect

exoplanets in the habitable zones of their stars using similar instruments, with the latter focusing on stars closer to Earth—TESS could benefit greatly from this approach. Additionally, the continued discovery of new candidate exoplanets by TESS further supports the potential for this method to enhance exoplanet classification. Moreover, the forthcoming PLATO mission[21], set to launch in 2026, aims to provide another vast data set for exoplanet research. This method could be instrumental in analyzing data from future large-scale transit surveys, making it a valuable tool for upcoming astronomical missions.

Data availability

All data generated or analyzed during this study are included in this article and its tables.

Received: 14 September 2024; Accepted: 15 April 2025
Published online: 02 May 2025

References

1. Brennan, P. *Why Do Scientists Search for Exoplanets? Here Are 7 Reasons*. Exoplanet Exploration. Retrieved February 3, 2024, from https://exoplanets.nasa.gov/news/1610/why-do-scientists-search-for-exoplanets-here-are-7-reasons/ (2019).
2. Jenkins, J. M. et al. Discovery and validation of Kepler-452b: A 1.6 R⊕ super earth exoplanet in the habitable zone of a G2 star. *Astron. J.* **150**(2), 56. https://doi.org/10.1088/0004-6256/150/2/56 (2015).
3. National Academies of Sciences, Engineering, and Medicine. *Exoplanet Science Strategy* (The National Academies Press, 2018).
4. NASA Exoplanet Archive. Exoplanet and candidate statistics. Retrieved January 23, 2025, from https://exoplanetarchive.ipac.caltech.edu/docs/counts_detail.html (n.d.).
5. Mayor, M. & Queloz, D. A Jupiter-mass companion to a solar-type star. *Nature* **378**(6555), 355–359. https://doi.org/10.1038/378355a0 (1995).
6. Tamayo, D. et al. Predicting the long-term stability of compact multiplanet systems. *Proc. Natl. Acad. Sci.* **117**(31), 18194–18205. https://doi.org/10.1073/pnas.2001258117 (2020).
7. Pinheiro, T. F. L. L., Sfair, R., & Ramon, G. Machine learning approach for mapping the stable orbits around planets. arXiv preprint arXiv:2412.04568. (2024).
8. Jin, Y., Yang, L. & Chiang, C.-E. Identifying exoplanets with machine learning methods: A preliminary study. *Int. J. Cybern. Inform.* **11**(1/2), 32–42 (2022).
9. Nath-Ranga, R., Absil, O., Christiaens, V. & Garvin, E. O. Machine learning for exoplanet detection in high-contrast spectroscopy. *Astron. Astrophys.* **689**, A142. https://doi.org/10.1051/0004-6361/202449150 (2024).
10. Borucki, W. J. et al. Kepler planet-detection mission: introduction and first results. *Science* **327**, 977–980. https://doi.org/10.1126/science.1185402 (2010).
11. NASA Exoplanet Archive. Kepler Object of Interest (KOI) table. Retrieved March 7, 2025, from https://exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbls&config=koi (n.d.).
12. Park, Y.-S., & Lek, S. Chapter 7—artificial neural networks: multilayer perceptron for ecological modeling. In *Developments in Environmental Modelling* (ed. Jørgense, S. E.), Vol. 28, 123–140. ISSN 0167-8892. ISBN 9780444636232. https://doi.org/10.1016/B978-0-444-63623-2.00007-4 https://www.sciencedirect.com/science/article/pii/B9780444636232000074 (Elsevier, 2016).
13. Chen, Y. et al. Chapter 2—Fundamentals of neural networks. In *AI Computing Systems* (eds Chen, Y. et al.) 17–51 https://doi.org/10.1016/B978-0-32-395399-3.00008-1 (2024).
14. Ahmed, Z., D'Amico, S., Hu, R., & Damiano, M. Exoplanet detection from starshade images using convolutional neural networks. SPIE. Retrieved from https://slab.stanford.edu/sites/g/files/sbiybj25201/files/media/file/ahmed_spie2023_submitted.pdf (2023).
15. Raj, R. & Kos, A. An improved human activity recognition technique based on convolutional neural network. *Sci. Rep.* **13**, 22581. https://doi.org/10.1038/s41598-023-49739-1 (2023).
16. NASA. *Data columns in Kepler Objects of Interest Table*. NASA Exoplanet Archive. Retrieved March 3, 2024, from https://exoplanetarchive.ipac.caltech.edu/docs/API_kepcandidate_columns.html (2021).
17. SciKit Learn. *1.17. Neural network models (supervised)—scikit-learn 1.4.1 documentation*. Scikit-learn. Retrieved March 3, 2024, from https://scikit-learn.org/stable/modules/neural_networks_supervised.html (n.d.).
18. Kingma, Diederik, P. & Ba, J. Adam: a method for stochastic optimization. CoRR abs/1412.6980, n. pag (2014).
19. SciKit Learn. *3.3. Metrics and scoring: quantifying the quality of predictions*. Scikit-learn. Retrieved March 3, 2024, from https://scikit-learn.org/stable/modules/model_evaluation.html#roc-metrics (n.d.).
20. Ricker, G. R. et al. The transiting exoplanet survey satellite. *J. Astron. Telescopes Instrum. Syst.* **1**(1), 014003. https://doi.org/10.1117/1.jatis.1.1.014003 (2014).
21. Rauer, H. et al. The PLATO 2.0 mission. *Exp. Astron.* **38**, 249–330. https://doi.org/10.1007/s10686-014-9383-4 (2014).

Acknowledgements

Graduate from Carnegie Mellon---Husni A. For mentoring and teaching me

Author contributions

Juliana Wang Wrote the paper, the code, and generated the table and graphs in the paper. Code was based off of Neural Network libraries. Taught and guided by mentor from Carnegie Mellon Husni A.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to J.W.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.