Comparative Analysis of Machine Learning and Neural Network Approaches for Exoplanet Identification

MSc Research Project
Artificial Intelligence

Buse Ay
Student ID: 23122633

School of Computing
National College of Ireland

Supervisor: Mayank Jain

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Buse Ay
Student ID:	23122633
Programme:	Artificial Intelligence
Year:	2024
Module:	MSc Research Project
Supervisor:	Mayank Jain
Submission Due Date:	12/08/2024
Project Title:	Comparative Analysis of Machine Learning and Neural Network Approaches for Exoplanet Identification
Word Count:	3754
Page Count:	12

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	16th September 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Comparative Analysis of Machine Learning and Neural Network Approaches for Exoplanet Identification

Buse Ay
23122633

Abstract

Several missions collect vast amounts of data from space every day. One of the purposes of that is to unravel the mystery of exoplanets. Due to the volume and the complexity of data, machine learning, and deep learning methods became popular for exoplanet identification. In this study, we used flux entries for multiple stars from Kepler Mission and applied a range of machine learning and deep learning algorithms, including K-Nearest Neighbors (KNN), Decision Tree, Logistic Regression, AdaBoost, XGBoost, Fully Connected Neural Networks, 1D Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM) network. For the evaluation accuracy, precision, recall, and F-1 score metrics were used. Results indicate that ensemble methods worked better on this specific data. AdaBoost and XGBoost, outperformed neural networks being more straightforward. The data used in this project is noisy, and because of that neural networks might learn noises instead of the flux patterns. The importance of model selection depending on the dataset for the identification of exoplanets is highlighted in this paper.

1 Introduction

The curiosity about the distant worlds has been a fascinating topic for astronomers and scientists for years. These distant worlds, planets outside of our solar system are called exoplanets, and the detection of exoplanets holds a promising future for human life and unraveling mysteries of space along with extraterrestrial life. Since the 1990s, the detection of the first exoplanet, research has grown exponentially with technological advancements. NASA's developments marked significant milestones in this field such as Kepler Space Telescope and TESS(Transitting Exoplanet Satellite). Both of these developments served with different datasets at different times. Recently, researchers focused on TESS datasets due to its wide research area but there are many challenges in extracting and understanding the data structure. The Kepler Space Telescope dataset is easier to understand. Its research area is also wide and still has many unknowns. Kepler Space Telescope besides being designed to detect Earth-sized planets, detected thousands of planets including the smallest, the closest, the most distant, and some of the most interesting planets (Shallue and Vanderburg (2018)). At the beginning of the mission, the identification of exoplanets, removing anomalies, and false positives was done by humans, there were multiple catalogs and no uniformity. So, midway through, scientists started using automated ways to advance their research (Shallue and Vanderburg (2018)). One

of the most known techniques is called the transit method. This method was used in the early days of the Kepler Mission for hand calculations, and also it is the backbone of most of the automated ways. If an exoplanet passes in front of its star, from a certain perspective, it should create periodic drops in the stellar light curves, which are the fluxes in the function of time, this is called a transit event and those periodic drops represent exoplanet (Iglesias Álvarez (2024)). This commonly used and effective method also comes with challenges. The signals often faint and they come with a lot of noise. These noises can be stellar variability, instrumental errors, and other anomalies (Jara-Maldonado et al. (2020)). Over the past years, with the contribution of advanced technologies, several Machine Learning and Deep Learning methods, reduce these noises in the data. Using these advanced technologies also helps the handling of vast amounts of data in a shorter time. Machine learning algorithms are mainly used for the classification of the signals whether they are exoplanets or non-exoplanets, they are trained to recognize patterns. Deep learning models such as Convolutional neural networks are also used for the classification but on time series images. Both methods developed highly accurate results (Jara-Maldonado et al. (2020)).
Every year, the Earth we live in is getting closer to being inhabitable. Because of that finding Earth-like exoplanets is crucial for the future of humanity. There are so many different methods to identify these exoplanets. In this project, we are answering the question of which method is more suitable for this task considering noisy astronomical datasets. We compare multiple machine learning and deep learning algorithms by their efficiency and classification performance. Our objective is to determine how well each method handles noisy flux variation datasets. Through this analysis, the project highlights the model selection based on the characteristics of the dataset. This paper is organized as follows. Related works for exoplanet detection and different methods in 2, details about the data and preprocessing steps along with methods choices in 3, description of machine learning algorithms such as K-NN, Decision Tree, Logistic Regression and, Convolutional Neural Network in 4, implementation of these methods in 5, results of the applications in 6, and conclusion and future works in 7.

2 Related Work

Methods of detecting exoplanets vary depending on the mission and data type. There are also a variety of machine learning and deep learning algorithms. Most of these approaches work successfully but mainly, for Kepler Mission, machine learning algorithms work better because the dataset is usually in CSV file format, and for TESS, deep learning algorithms work better because datasets are in image format. A survey done by Jara-Maldonado et al. (2020) summarizes the main approaches and they consider both datasets. They also explain exoplanets and their detection methods using light curves and radial velocity curves. They generalized the detection process into four different steps, these are light curve acquisition, light curve preprocessing, transit signal detection, and transit signal identification. Also, they introduced different methods for different steps. These methods include Least Square, Bayesian Approaches, Match Filters, Decision Trees, Support Vector Machines, Convolutional Neural Networks, and Astronet(deep learning architecture for exoplanet detection). They mentioned noticeable advances in this field but, also emphasized that the noisy features are still present. They suggested that the ideal model must analyze feeble signals(Jara-Maldonado et al. (2020)).

While some researchers focus on the Kepler data, others focus on the TESS data. One of the studies done by Bhamare et al. (2021) focused on the Kepler data and analyzed Kepler Objects of Interest, meaning that the target displays at least one transit-like sequence. Their dataset has three main labels 'False Positives', 'Confirmed', and 'Candidate'. Like them, most of the other researchers do not consider 'Candidate' labeled data due to their unknown characteristics. In their study, they used a Support Vector Machine, Random Forest, AdaBoost, and Feed-Forward Neural Network methods. All methods achieved more than 97%. F-1 Score, AdaBoost was their best-performing method. They used Principal Component Analysis(PCA) to maximize the variance and retain an uncorrelated feature set, this approach was used in this paper as well due to the uncorrelated characteristic of the dataset. They admit that the nature of the dataset and unknown reasons for missing values challenged them. While they achieved high performance, the study needs a deeper analysis of algorithm performance and interpretability. The exclusion of 'Candidate' data could lead to potential biases. Another study that focused on the Kepler data is written by Khan and Dixit (2020). They also centralized their research around Kepler's Object of Interest. They implemented deep neural networks and Support Vector Machine conjunction and achieved 98%. F1-Score. Like Bhamare et al. (2021), Khan and Dixit (2020) used Principal Component Analysis to deal with feature variance. Besides their good results, they also underline that noises remain. However, they did not consider the computational costs of models and evaluation of false positives. One of the important works done by Shallue and Vanderburg (2018) reduced the noise on Kepler data as much as possible and discovered two new exoplanets by their method. They used a deep convolutional neural network and achieved 98.8%. accuracy. They trained their model on human-classified Kepler TCEs(Threshold Crossing Events), a list of planet candidates culled by humans, and they tested their models on a new set of candidate signals. Instead of working on KOIs(Kepler Object of Interest), they worked on light curves and used their convolutional neural network architecture on them, this architecture is called Astronet. They implement three different neural networks Linear, Fully Connected, and Convolutional networks. All achieved more than 90%. accuracy. They mentioned that they are planning to improve their model to identify failure modes, they discover in their study. However Astronet is a successful method, it is known as the 'black box' model as well, which limits the transparency and interpretability of the results. Following their study, Dattilo et al. (2019) used Astronet architecture on Kepler extended K2 mission. They modified the neural network for the K2 mission as Astronet-K2 and it was also achieved 98%. accuracy. One of the reasons for the modification method was the different time span of the K2 mission. Their main challenge was labeling the data due to missing TCEs for the K2 Mission. They have also identified two new exoplanets and they are planning to apply their model on larger datasets. One of the drawbacks of their model, it requires a large and well-labeled dataset, which is not feasible.

Besides using Kepler data, many researchers used the TESS data. Another study that follows the Shallue and Vanderburg (2018) method was done by Yu et al. (2019). They used Astronet architecture on the TESS light curves. Like Dattilo et al. (2019), they also made changes to the original model and they achieved 97%. precision and 97.4%. accuracy. They also admit that their research relies on human-culled catalogs. Yu et al. (2019) used Triage and Vetting mode for their Astronet convolutional network approach. Their triage achieved better results than their vetting. As a challenge, they also accepted the errors in human-culled catalogs, these might lead the model to wrong assumptions. Their future scope is to adapt their model for multi-class classification. Also, Iglesias Álvarez (2024) used TESS data and they implemented two 1D Convolutional Neural Networks. The first model works on light curves and the second model works on phase-folded light curves. They were more focused on the mathematical side of the application and error calculations. However, they could emphasize more on challenges like noise, missing values, and bias from human-labeled catalogs.

There are new approaches for identifying exoplanets. One of the studies done by Valizadegan et al. (2022), implemented a model called ExoMiner on Kepler and TESS data and they validated 301 new exoplanets. They used Data Validation(DV) files as input and applied deep neural networks to them. Another new approach is done by Liao et al. (2024) and implements a CNN model based on Inception-v3 for light curve classification. They achieved 95%. accuracy. They used Wavelet Multiresolutions which decompose signals in different frequency components. They reduced noise successfully but they faced with sensitivity of algorithms to measure errors.

3 Methodology

The research methodology includes data collection, data preprocessing, model training, and evaluation. Data is collected from Kaggle (WDelta (2017)). It originally belonged to NASA's Kepler Campaign-3 and describes flux changes. It includes training and test datasets. The train set has 5087 rows and 3198 columns, rows represent stars and columns are measured fluxes at each time interval. Both sets have binary labels as 2 for exoplanet stars and 1 for non-exoplanet stars. These labels changed to 1 and 0 to apply classification models easier. Distribution of labels shown in 1. After checking the distribution, missing values are checked as well and there are no missing values. As a next step, a correlation map was plotted since the fluxes are independent of each other, The correlation matrix did not give good results. To understand the data better, in Figure 2 flux variations are plotted. As seen, exoplanet fluxes have more periodic dips compared to the non-exoplanet flux variations. Still, there are fluctuations in the data, these are caused by noises or instrumental errors.
After visualizing, data normalization was performed by using `sklearn` library to scale the range. Because data has a large number of features, principal component analysis (PCA) is used on high-dimensional data as Khan and Dixit (2020) and Bhamare et al. (2021) did. It is a common technique to transform large variables to smaller ones while saving the informationKhan and Dixit (2020). As seen in Figure 1, data is unbalanced because of that, the Synthetic Minority Over-sampling Technique (SMOTE) is used. This technique helps the balance label disturbance by generating samples for the exoplanet class.
With normalized and balanced data, different machine learning algorithms are trained. Such as K-Nearest Neighbors (KNN), Decision Tree Classifier, Logistic Regression, AdaBoost Classifier, XGBoost, and Neural Networks. Four different neural network designs were applied. These are Fully Connected Networks with SELU Activation and L2 Regularization, with Swish Activation and Adam Optimizer, 1D Convolutional Neural Networks, and Long Short-Term Memory (LSTM) Networks. Training all these models and comparing the results is important to evaluate and see which models work better on the flux dataset. For the evaluation of the results, accuracy, precision, recall, and F-1 score

is used, for the machine learning algorithms ROC Curve graph plotted.

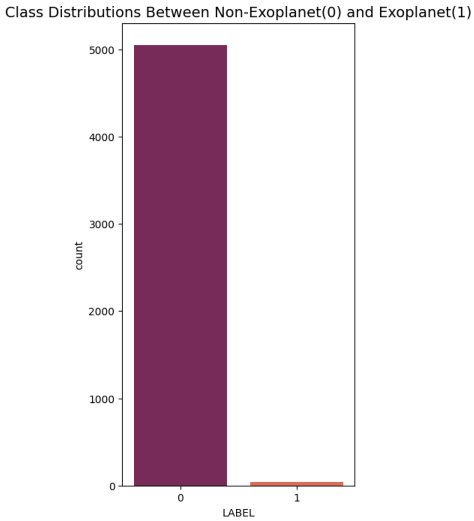

Figure 1: Distubition of labels

4 Design Specification

The pipeline of the project includes steps of data preprocessing, dimensional reduction using PCA, data balancing with SMOTE, and model training. The design allows the comparison of models with a standardized evaluation matrix. Models trained are K-Nearest Neighbors (KNN), Decision Tree Classifier, Logistic Regression, AdaBoost Classifier, XGBoost, and Neural Networks such as Fully Connected Networks with SELU Activation and L2 Regularization, with Swish Activation and Adam Optimizer, 1D Convolutional Neural Networks, and Long Short-Term Memory (LSTM) Networks.

K-Nearest Neighbors (KNN) is a method used for classification and regression, with given vectors and it identifies k-nearest neighbors (Alzubi et al. (2018)). It is simple to implement because there is no training phase, all happens during the prediction. It is sensitive to the choice of k and not effective for high dimensional spaces.

Decision Tree is also a method used for classification regression. It uses tree-like structures to split the data into subsets, each node represents decisions, each branch represents outcomes and each leaf represents labels(Alzubi et al. (2018)). It can be used for both categorical and numerical data however it is not stable, and small changes can cause biases.

Logistic Regression is commonly used for binary classifications by using logistic functions like sigmoid (Alzubi et al. (2018)). This model is sensitive the outliers so it needs detailed preprocessing steps. Usually gives probabilistic outputs and works better with linearly separable data.

AdaBoost and **XGBoost** are ensemble techniques of decision tree. Adaptive Boosting (AdaBoost) combines multiple decision stumps (decision trees with single split) to

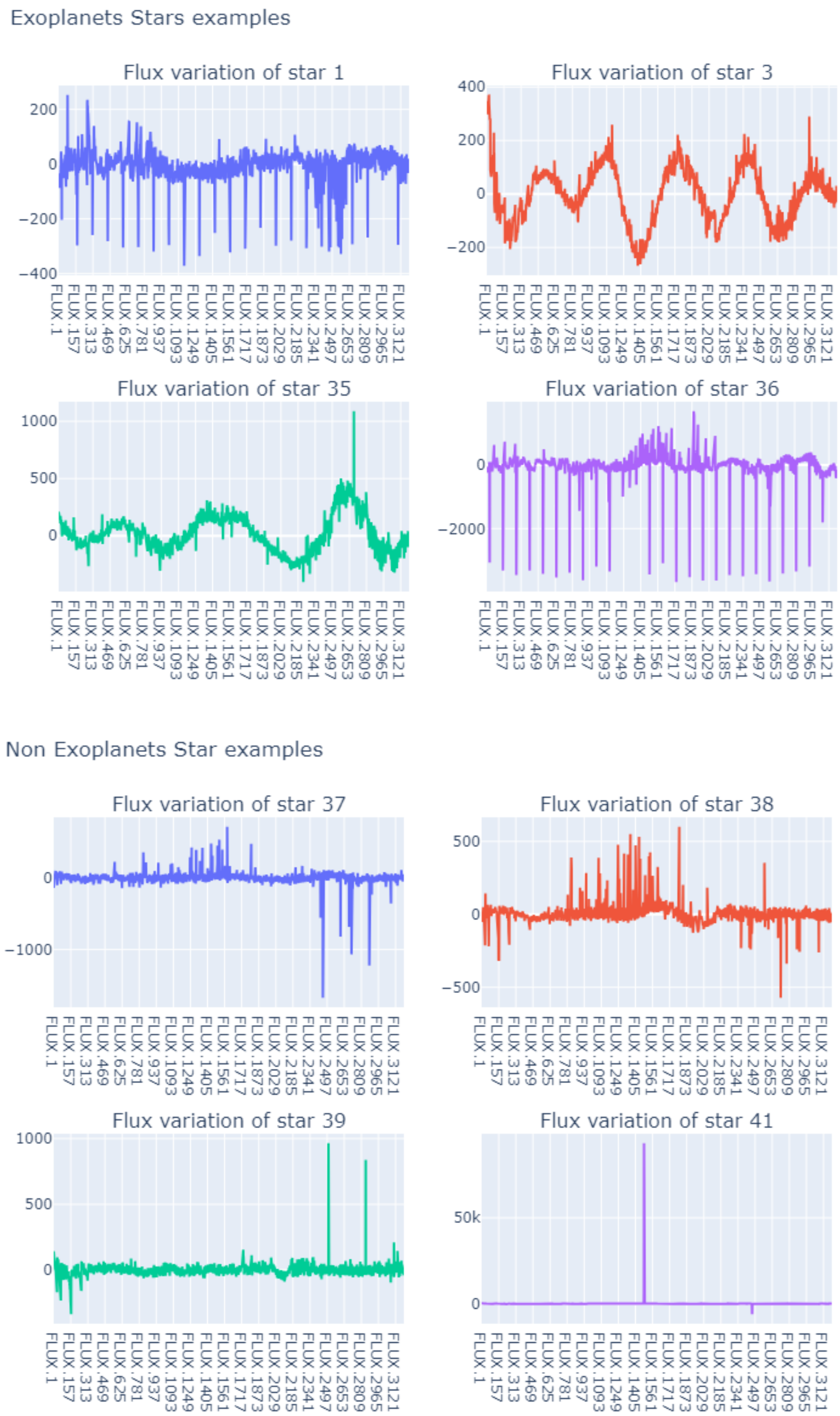

Figure 2: Exoplanet and Non Exoplanet Flux Variations

create a strong classifier(Azmi and Baliga (2020)). Each time it focused on the mistakes and combined the predictions using a weighted majority vote. Extreme Gradient Boosting (XGBoost) builds sequential decision trees, and each tree corrects previous ones(Azmi and Baliga (2020)).

Neural Networks models are inspired by the human brain with consistent layers. Their architecture mainly contains an input layer, one or more hidden layers, and an output layer. Besides these layers, there are activation functions, regularizers, optimizers, etc. Neural networks are highly effective, flexible, and capable of handling complex tasks(Shallue and Vanderburg (2018)). It is mainly used for classification problems by detecting patterns. *Fully Connected Neural Networks* also known as Dense Network is an architecture that has hierarchical neurons, so the outputs from one layer are inputs to the next(Shallue and Vanderburg (2018)). One of the designs for a fully connected neural network used in this paper consists of fully connected layers, where each layer uses the SELU activation function and L2 regularization applied to the weights. Another one consists of fully connected layers with Swish activation functions and a model trained by using an Adam optimizer. *1D Convolutional Neural Networks* are specialized for sequential data types. 1D CNN consists of convolutional layers with pooling and fully connected layers(Shallue and Vanderburg (2018)). It learns local features once and this reduces memory usage and computational power(Shallue and Vanderburg (2018)). These methods were chosen because they are optimal for sequential data like flux values. If there is an exoplanet, flux needs sequential and repetitive in a certain amount of time and depth. Because flux entries are measured at each time interval even though it is not time-series data Long Short-Term Memory (LSTM) Network also trained. LSTMs are a type of recurrent neural network to solve vanishing gradient problems and they are usually used for time-series datasets.

The chosen machine learning algorithms are well-suited for classification problems. Traditional ML models provide insight for the classification process of our data. Logistic regression was chosen as a baseline model. While KNN provides a better understanding of local relationships the Decision Tree captures the non-linear relations. Ensemble methods such as AdaBoost and XGBoost bring predictive power for imbalanced classes. Fully Connected Networks with SELU and Swish activations minimize the overfitting on high-dimensional flux data and 1D CNNs detect features like dips in flux which leads us to the identification of exoplanets.

5 Implementation

The implementation of the project was carried out in the Google Colab environment by using Python coding language. Google Colab environment provides GPU acceleration and other necessary computational resources, which are beneficial for training machine learning and neural network models. Main Python libraries used in this project include,

- `Numpy` and `Pandas` for data handling and manipulations
- `Seaborn` and `Matplot` for visualization
- `Scikit-learn` for data preprocessing such as normalization and PCA, and machine learning applications

- `TensorFlow` and `Keras` for neural network applications
- `Imbalanced-learn` for using SMOTE to balance the dataset.

The train and test dataset was imported into the Google Colab environment. After visualization for a better understanding of data, missing values are checked and dealt with. Data normalized and dimensionally reduced by using PCA. Imbalanced data handling applied by using SMOTE. After the preprocessing steps are over, for machine learning applications data is split into `x_train,y_train,x_test,y_test` by using test size 0.33, and for neural network application data is split by using 0.2 test size. Except for the XGBoost other machine learning models are implemented with `Scikit-learn` library and XGBoost is implemented with `XGBoost` library. Neural networks are applied by using `TensorFlow/Keras` libraries. The first application of neural networks uses three fully connected layers with 300,200 and 100 neurons each utilizing the SELU activation function. To prevent overfitting L2 regularization was applied with a regularization factor of 0.01. The last layer has 2 neurons with softmax activation function for distribution for exoplanets and nonexoplanets. The second application of the neural network has the same three layers with the Swish activation function and He Normal initializer. And the final layer is the same as the first application. For this time Adam optimizer tested with a learning rate of 0.001 to prevent instability. The third application employs a 1D Convolutional Neural Network. It consists of two 1D convolutional layers followed by max pooling layers. While the first 1D convolutional layer applies 64 filters, the second convolutional layer applies 128 filters. Both use ReLU activation. The output of these layers is flattened and followed by two dense layers. The fourth and last application of neural networks is the Long Short-Term Memory (LSTM) network. It consists of two LSTM layers with units 64 and 128 followed by dense layers. It also uses Adam Optimizer. All neural networks trained for 50 epochs and learning curves for accuracy and loss are plotted. All classification reports include precision, recall, and F1 scores.

After training, model evaluation is done by using a separate test set, and accuracy, precision, recall, and F1 scores are listed. Additionally, confusion matrixes were generated and for machine learning algorithms ROC Curves were plotted. For evaluation `Scikit-learn` and `TensorFlow/Keras` libraries are used and for visualization `Seaborn/Matplot` libraries are used.

6 Evaluation

The evaluation of the project focused on explaining the performances of different models trained during the implementation phase. The models were evaluated on a separate test set in terms of accuracy, precision, recall, and F1 score. Accuracy measures the ratio of correctly classified cases for both true positives and true negatives out of a total number of cases. Precision, Positive Predictive Value, measures the ratio of true positive cases out of all positive cases. High precision means the model predicts positive classes correctly. Recall, that the sensitivity, measures the ratio of true positives to actual positives which is the sum of true positives and false negatives. High recall means the model successfully predicts positive cases. F1-Score is the harmonic mean of Precision and Recall, a single metric represents precision and recallJara-Maldonado et al. (2020). High F1-Score means balanced model, for the imbalanced data this metric is crucial. Confusion matrixes are

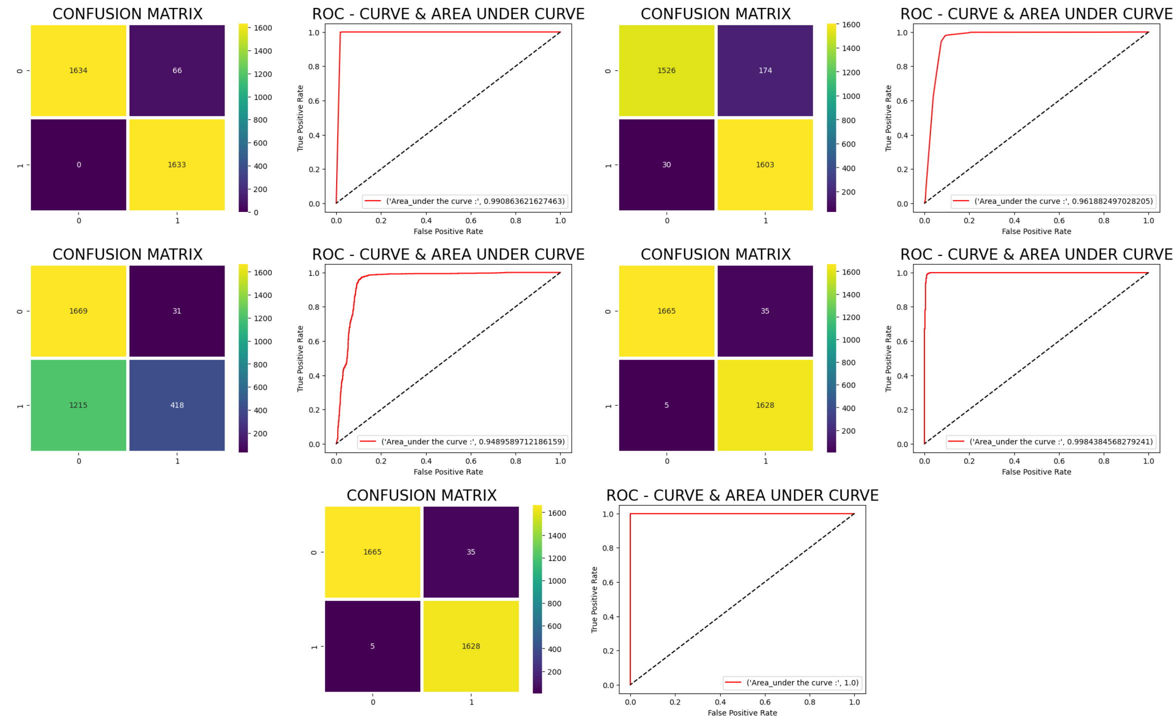

Figure 3: Confusion Matrixes and ROC Curves for Machine Learning Models (a) KNN (b) Decision Tree (c) Logistic Regression (d) AdaBoost (e) XGBoost

tables that describe the performance of the model including the counts of true positives, true negatives, false positives, and false negatives. The Roc Curves plots represent the ability of binary classifier systems. If the Area Under the ROC Curve is 1 it means the model is a perfect classifier. Confusion Matrixes and ROC Curves of machine learning algorithms shown in Figure 3.

Results are shown in Table1 and the distribution of results is shown in Figure 4. The KNN, AdaBoost, and XGBoost achieved the highest accuracy, and AdaBoost and XGBoost achieved the highest precision showing that these models are successful at minimizing false positives. In contrast, neural network models have lower precision. XGBoost achieved the highest recall and shows that it is effective for identifying true positives. Again, AdaBoost and XGBoost achieved the highest F1-Score, which means overall performance in balancing precision and recall is successful. The evaluation indicates that AdaBoost and XGBoost outperformed the other models. These ensemble methods are effective at identifying exoplanets for this dataset.

6.1 Discussion

In this project, machine learning and deep learning algorithms were applied to identify exoplanets by using the flux entries dataset of the Kepler Mission. These models are K-Nearest Neighbors (KNN), Decision Tree, and Logistic Regression to more advanced techniques like AdaBoost, XGBoost, Fully Connected Neural Networks with SELU and Swish activations, 1D Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM) networks. Evaluation metrics showed us ensemble methods like AdaBoost and XGBoost achieve better results in general. Deep learning models showed high accuracy but their F-1 Scores were lower than ensemble methods. Traditional machine learning

Model	Accuracy	Precision	Recall	F1-Score
KNN	0.98	0.98	0.98	0.98
Decision Tree	0.93	0.94	0.94	0.94
Logistic Regression	0.62	0.75	0.62	0.56
AdaBoost	0.98	0.99	0.99	0.99
XGBoost	0.98	0.99	0.99	0.99
Fully Connected (SELU)	0.63	0.50	0.52	0.60
Fully Connected (Adam Optimizer)	0.88	0.50	0.50	0.50
1D CNN	0.92	0.50	0.49	0.49
LSTM	0.87	0.88	0.87	0.87

Table 1: Results

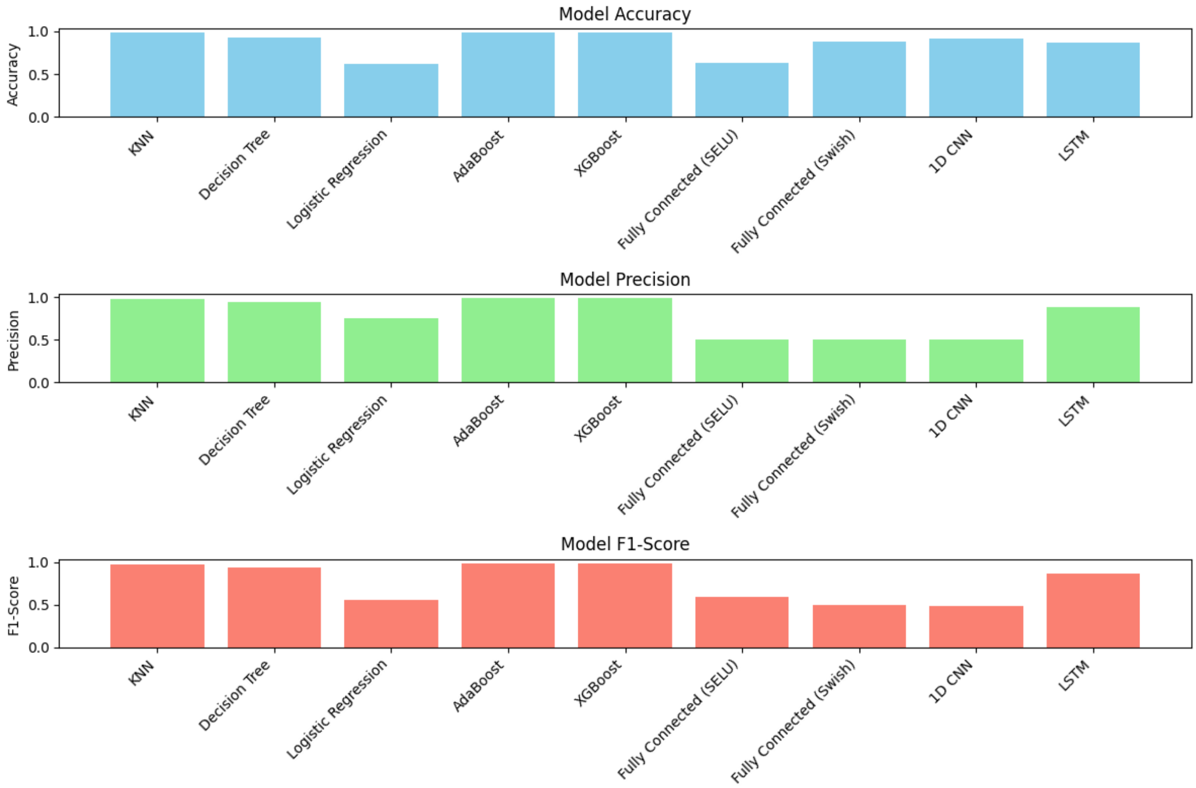

Figure 4: Result Disturbition Graph

models also achieved good accuracy but these models might struggle with the complexities of identifying exoplanets for other datasets. Also, the ROC curves and AUC values show ensemble methods and deep-learning models performed better overall.
The reason neural networks did not outperform ensemble methods might be the complexity of the dataset. If the data set is not large enough, the model might overfit and start learning noises. In our case noises are more than the patterns, so this highly might be the reason ensemble models worked better than neural networks for this application. Also, ensemble methods are known for their performance in handling structured data, our data has over 3000 flux entries for each star, which might be the other reason. Neural networks are very sensitive so the quality and size of the data for this project were not sufficient for neural network applications. On the other side, with the ability to reduce overfitting and combine multiple models, the ensemble model was suited for this dataset.

7 Conclusion and Future Work

This study showed machine learning and deep learning algorithms, especially ensemble methods effective for exoplanet identification. While traditional machine learning algorithms are used for baseline, ensemble models such as AdaBoost and XGBoost and deep learning architectures such as 1D CNN and LSTM are compared. While neural networks are powerful approaches, this study was outperformed by the ensemble model due to the nature of the dataset. Advanced activation functions and deep architecture might cause overfitting on not sufficiently large or diverse datasets and might cause learning noise. Ensemble methods are more straightforward in terms of the training process and hyperparameter tuning. At the end of this project, the ensemble models performed better than the neural networks due to the nature of the data. Future scope is using more complex datasets and different optimization techniques to evaluate the model performance from different perspectives. Another area of exploration could be employing a hybrid model that combines neural networks and ensemble methods for more accurate exoplanet identification.

References

Alzubi, J., Nayyar, A. and Kumar, A. (2018). Machine learning from theory to algorithms: An overview, *Journal of Physics: Conference Series* **1142**(1): 012012.
URL: *https://dx.doi.org/10.1088/1742-6596/1142/1/012012*

Azmi, S. S. and Baliga, S. (2020). An overview of boosting decision tree algorithms utilizing adaboost and xgboost boosting strategies, *Int. Res. J. Eng. Technol* **7**(5): 6867–6870.

Bhamare, A. R., Baral, A. and Agarwal, S. (2021). Analysis of kepler objects of interest using machine learning for exoplanet identification, *2021 International Conference on Intelligent Technologies (CONIT)*, pp. 1–8.

Dattilo, A., Vanderburg, A., Shallue, C. J., Mayo, A. W., Berlind, P., Bieryla, A., Calkins, M. L., Esquerdo, G. A., Everett, M. E., Howell, S. B., Latham, D. W., Scott, N. J. and Yu, L. (2019). Identifying exoplanets with deep learning. ii. two new super-earths

uncovered by a neural network in k2 data, *The Astronomical Journal* **157**(5): 169.
URL: *https://dx.doi.org/10.3847/1538-3881/ab0e12*

Iglesias Álvarez, S.; Díez Alonso, E. S. R. M. R. R. J. P. F. S. d. C. J. F. (2024). Computing transiting exoplanet parameters with 1d convolutional neural networks, *Axioms* **13**(83).

Jara-Maldonado, M., Alarcon-Aquino, V., Rosas-Romero, R., Starostenko, O. and Ramirez-Cortes, J. (2020). Transiting exoplanet discovery using machine learning techniques: A survey, *Earth Science Informatics* .

Khan, M. A. and Dixit, M. (2020). Discovering exoplanets in deep space using deep learning algorithms, *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)*, pp. 441–447.

Liao, H., Ren, G., Chen, X., Li, Y. and Li, G. (2024). Identifying exoplanet candidates using waveceptionnet, *The Astronomical Journal* **167**: 180.

Shallue, C. J. and Vanderburg, A. (2018). Identifying exoplanets with deep learning: A five-planet resonant chain around kepler-80 and an eighth planet around kepler-90, *The Astronomical Journal* **155**(2).

Valizadegan, H., Martinho, M. J. S., Wilkens, L. S., Jenkins, J. M., Smith, J. C., Caldwell, D. A., Twicken, J. D., Gerum, P. C. L., Walia, N., Hausknecht, K., Lubin, N. Y., Bryson, S. T. and Oza, N. C. (2022). Exominer: A highly accurate and explainable deep learning classifier that validates 301 new exoplanets, *The Astrophysical Journal* **926**(2): 120.
URL: *http://dx.doi.org/10.3847/1538-4357/ac4399*

WDelta (2017). Exoplanet hunting in deep space.
URL: *https://www.kaggle.com/datasets/keplersmachines/kepler-labelled-time-series-data/data*

Yu, L., Vanderburg, A., Huang, C., Shallue, C. J., Crossfield, I. J. M., Gaudi, B. S., Daylan, T., Dattilo, A., Armstrong, D. J., Ricker, G. R., Vanderspek, R. K., Latham, D. W., Seager, S., Dittmann, J., Doty, J. P., Glidden, A. and Quinn, S. N. (2019). Identifying exoplanets with deep learning. iii. automated triage and vetting of tess candidates, *The Astronomical Journal* **158**(1): 25.
URL: *https://dx.doi.org/10.3847/1538-3881/ab21d6*