International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Feature Engineering for False Positive Exoplanet Prediction: A Deep Learning Approach

Nidhi Shakhapur, Ravin D., Chetan Shiraguppi, Dr. Sathya K.

Submitted: 07/02/2024 **Revised:** 14/03/2024 **Accepted:** 22/03/2024

Abstract*:* The growing number of satellites has improved our understanding of exoplanets, but it has also increased false positive detections. These errors can mislead research and allocation of resources. To address this, we introduce ArtAe, an AI model that employs Artificial Neural Networks and AutoEncoders to validate exoplanet data. ArtAe processes Kepler and TESS datasets, achieving 93.67% and 92.10% accuracy respectively in distinguishing genuine exoplanets from false positives. Moreover, this model has unique algorithm that reduces overfitting of the model. It also lowers dataset dimensionality, saving time and resources. This accuracy aids in informed resource allocation for future studies and enables automated, accurate data validation and analysis.

Keywords: *Artificial Neural Networks, Deep Learning, Exoplanet, Stellar Parameters, Transit Properties*

1. Introduction

Proliferation of satellites has revolutionized our understanding of exoplanets, marking a pivotal moment in astronomical exploration. However, this surge in satellite data acquisition has led to an alarming rise in the incidence of false positive exoplanet detections. These erroneous readings have the potential to distort our conclusions regarding exoplanet existence and characteristics, with consequences for the allocation of vital resources, including time and funding, for further study and exploration. In response to this pressing challenge, we introduce ArtAe, an advanced Artificial Intelligence Model. By harnessing the capabilities of Artificial Neural Networks and AutoEncoders, ArtAe offers a robust solution for meticulously validating expansive exoplanet datasets. With a remarkable accuracy rate of 93.67% and 92.10% accuracy for Kepler and TESS datasets respectively, it effectively distinguishes authentic exoplanet candidates from false positive detections. Moreover, ArtAe optimizes data processing and analysis by reducing dataset dimensionality, resulting in resource savings in terms of time, computational power, and storage space. This heightened precision in exoplanet data promises to inform future research endeavours and resource allocation based on dependable information, thereby enhancing the prioritization of scientific pursuits. Additionally, ArtAe's versatile algorithms can be autonomously trained to validate and analyse vast datasets, markedly diminishing the need for manual intervention while elevating result accuracy.

[1] *Department of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India. Email: nidhi.shakhapur2021@vitstudent.ac.in*

[2] *Department of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India. Email: ravin.d2021@vitstudent.ac.in*

[3] *Department of Information Technology, Vellore Institute of Technology, Vellore, India. Email: chetan.shiraguppi2020@vitstudent.ac.in*

[4] *Department of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India. Email: sathya.k.@vit.ac.in (*Corresponding Author)*

A. Satellite Data Handling

• About satellite data handling for exoplanets

Satellite data handling for exoplanets involves the collection, processing, and analysis of data from various space-based observatories and telescopes. The data includes information on the light spectra, atmospheric compositions, and other features of exoplanets. [6] This information is used to study the physical and atmospheric conditions of these distant worlds, with the goal of understanding their potential for habitability and characterizing their environments. The data handling process involves the use of advanced algorithms, data reduction techniques, and machine learning models to extract meaningful insights from the vast amounts of data collected by satellites [2]. The ultimate goal is to improve our understanding of the formation and evolution of exoplanets and provide insights into the possibilities of life beyond our solar system.

• About False Positive Result in Exoplanet

False positive results in exoplanet detection refer to situations where a signal is interpreted as evidence of a planet, but it is later discovered that the signal is not actually due to a planet [3]**.** This can occur due to various reasons such as contamination from other sources, instrumental effects, or errors in data analysis. The manual work involved in removing false positive results in exoplanet detection involves a thorough examination of the data obtained from the observations to eliminate any signals that do not correspond to actual exoplanets [4]**.** Scientists use data, such as radial velocity or transit parameters, to confirm the presence of an exoplanet [5]**.** This process requires a lot of time, effort, and expertise, and is essential for obtaining accurate results in exoplanet detection.

B. Application of Deep Learning in Space Technologies

• AI in satellite Data handling

Artificial Intelligence (AI) is increasingly being used to improve the accuracy and efficiency [19] of satellite data handling in the detection of exoplanets. AI algorithms, such as machine learning and deep learning, can be trained to identify patterns and anomalies in large amounts of data, reducing the time and manual effort required for exoplanet detection [1]. AI can also help in reducing false positive results by providing more accurate and reliable data. By automating various tasks in satellite data handling, such as image processing and feature extraction, AI can help scientists make better and faster decisions in exoplanet research.

• AI to reduce manual work

Artificial Intelligence (AI) can be used to automate the calculation of transit parameters in satellite data handling for exoplanet detection [18]. Traditional methods often require a lot of manual work, including the visual inspection of light curves and the use of complex mathematical models to determine the transit parameters [7]. AI techniques, such as machine learning algorithms, can be trained on large datasets to accurately and efficiently calculate the transit parameters. The use of AI can significantly reduce the manual workload and improve the speed and accuracy of transit parameter calculation, leading to more efficient exoplanet detection and characterization [6].

C. Deep Learning Models:

• Artificial Neural Networks (ANN)

Artificial Neural Networks (ANNs) are a type of machine learning model inspired by the structure and function of the human brain [12]. They consist of interconnected nodes, also known as artificial neurons, which process information and make decisions based on that information [9]. ANNs can be trained to recognize patterns in data, classify information, and perform various other tasks [11]. Their ability to learn from examples and make decisions based on that learning makes them widely applicable across a variety of industries, including finance, healthcare, and other marketing [8]. Despite their popularity, ANNs can be computationally expensive and complex to design, requiring a trade-off between the accuracy and efficiency [10].

• Auto Encoders:

Auto Encoders are a type of neural network architecture designed to learn a compact representation, or encoding, of input data through a process of encoding and decoding. They consist of two main components: an encoder that maps the input data to a lower-dimensional representation, and a decoder that maps the encoding back to the original input space [13]. Auto Encoders are commonly used for various tasks that includes dimensionality reduction, anomaly detection, and generative modelling [14]. They can also be combined with other neural network models to improve their performance in various applications [15]. Despite their effectiveness, Auto Encoders can suffer from overfitting and convergence issues, particularly in large and complex datasets [15].

• About ArtAe

ArtAe stands for Artificial Neural Networks powered by Auto Encoders. The Auto Encoders part reduces the dimensions of the high dimensional dataset and the results are predicted by Artificial Neural Networks.

In the paper [16] *Exoplanet detection using machine learning, Abhishek Malik, et al.*, implemented a classical Machine Learning Models from *TSFRESH* and they used the extracted features from their model to train a gradient boosting classifier using the machine learning tool *LIGHTBGM*. ArtAe is a deep learning model which uses TensorFlow and Keras.

Also in the paper [16], the authors have implemented a Deep Convolutional Neural Network to predict whether a given signal is a transiting exoplanet or a false positive caused by astrophysical or instrumental phenomena. But, The ArtAe Model integrates Auto Encoders with Artificial Neural Networks and can eliminate false positives using only transit properties.

In comparison to classical machine learning models, ArtAe offers improved accuracy and hyper-parameters using TensorFlow and Keras which gives optimised results than the former.

2. Literature Review

We have analyzed and reviewed the following papers that use Machine Learning and Deep Learning models to classify the false positivity of the exoplanets.

Ref.*	Year Published	Catalog	ML/DL Method	Performance
[18]	2015	Kepler TCEs	Random Forest Classifier	PC (0.971/2.9%), AFP (0.976/2.4%), NTP (0.968/3.2%)
[15]	2018	Autovetter	Convolutional Neural Network (CNN)	Recall: 95%, Accuracy: 90%, Precision: 96%
[20]	2018	Autovetter	CNN	Accuracy: 97.5%, Precision: 95.5%
[21]	2019	TESS Candidates	Modified Astronet	Triage: Precision 97.0%, Accuracy

				97.4%; Vetting: Precision 69.3%, Accuracy 97.8%
[22]	2019	Kepler KOI	Random Forest Classifier	Accuracy: 98.96%, Precision: 99.55%, Recall: 97.21%
[23]	2019	Synthetic Data	CNN (2D Phase Folding)	Models with folding: Accuracy > 98%, Models without folding: Accuracy ≈ 85%
[16]	2020	Kepler & TESS Data	LightGBM with feature extraction	Kepler AUC: 94.8% accuracy, 96% recall; TESS Accuracy: 98%, Recall: 82%
[24]	2023	TESS Data	CNN Model with Transfer Learning (TL)	Accuracy: 87%
[26]	2022	Kepler KOI	TSFRESH & LIGHTGBM (Gradient Boosting)	Accuracy: 96%
[27]	2022	Kepler & TESS Data	ExoMiner (Proposed Deep Learning Classifier)	Accuracy: 93.6%
[28]	2023	TESS Data	Astronet-Triage-v2	Recall: 99.6%, Precision: 75.7%

Ref.*- References cited

While these models have demonstrated commendable accuracy and precision, they may be susceptible to overfitting since they lack the incorporation of algorithms designed to mitigate this issue. Overfitting is a phenomenon in which the machine excessively learns from the data to achieve a high accuracy rate, but the outcomes may not be entirely accurate. In contrast, our integrated deep learning model addresses this concern by incorporating a unique algorithm. Although it may yield slightly lower accuracy compared to certain models, it consistently produces precise and correct results, making it a more dependable choice.

3. Methodology

In this section, we delineate the methodologies employed in our study to enhance predictive accuracy. We begin with data pre-processing, focusing on the allocation of training and testing sets to facilitate model learning. Subsequently, we delve into dimensionality reduction utilizing Autoencoders, a pivotal step in managing high-dimensional datasets. This is followed by an exploration of the architecture and activation functions of Artificial Neural Networks, which constitute the core of our predictive model. Additionally, we discuss the optimizer, loss function, and metrics employed in training the Artificial Neural Network. Finally, we touch upon the key mathematical calculations underpinning our deep learning model.

Abbreviations and Acronyms:

ADAM	Adaptive Moment Estimation
AI	Artificial Intelligence
ANN	Artificial Neural Networks
CPU	Central Processing Unit
DL	Deep Learning
GPU	Graphical Processing Unit
KOI	Kepler Object of Interest
ML	Machine Learning
PCA	Principal Component analysis
ReLU	Rectified linear activation function
TPU	Tensor Processing Unit

A. Theory and Calculations:

3.1: Mathematics of Artificial Neural Network

These equations provide a high-level overview of the mathematics behind designing the architecture of the ANN.

Matrix multiplications:

In ANNs, weights are represented as matrices and input data as vectors. The dot product of these matrices and vectors represents the weighted sum of inputs that are used to compute the activations. The equation for matrix multiplication is given by:

$$C = A \,.\, B$$

where A and B are matrices, and C is the result of the matrix multiplication.

Vector operations:

ANNs use vector operations such as dot product, element-wise multiplication, and addition to compute activations and gradients during the forward and backward propagation steps. The dot product of two vectors is given by:

$$C = \overrightarrow{A} \,.\, \overrightarrow{B}$$

where C is the dot product of vectors A and B.

Matrix-vector multiplication:

In ANNs, the dot product of a weight matrix and an input vector is used to compute the activations. The equation for matrix-vector multiplication is given by:

$$\overrightarrow{c} = A \,.\, \overrightarrow{v}$$

where c is the result vector, A is the weight matrix, and v is the input vector.

Gradient Descent Equation:

This equation is used to update the model's parameters and minimize the loss function. The equation is as follows:

$$\theta = \theta - \eta \nabla \theta J(\theta)$$

where,

θ represents the model parameters

η represents the learning rate

J(θ) represents the loss function.

∇θ J(θ) represents the gradient of the loss function with respect to the model parameters.

Neural Networks specifically use Stochastic Gradient Descent, which is an extension of Gradient Descent that uses only a random subset of the training data at each iteration. The update equation for SGD is given by:

$$\theta = \theta - \alpha \cdot \nabla \theta J(\theta i)$$

where θi is a random sample from the training data.

Chain Rule of Calculus:

This rule is used to calculate the gradient of the loss function with respect to the model parameters. The chain rule is as follows:

$$\frac{d}{dx}[f(g(x))] = f'(g(x)) \cdot g'(x)$$

where $f(x)$ and $g(x)$ represent any two functions, and $f'(x)$ and $g'(x)$ represent the derivatives of $f(x)$ and $g(x)$ respectively.

The Backpropagation Algorithm:

The differentiation part of the backpropagation algorithm calculates the gradient of the loss function with respect to the model parameters using the chain rule. The gradient of the loss function with respect to the weights W1 and W2 can be calculated as follows:

$$\frac{dL}{dW_1} = \frac{dL}{dy} \cdot \frac{dy}{dW_1} \cdot \frac{dL}{dW_2} = \frac{dL}{dy} \cdot \frac{dy}{dW_2}$$

where dL/dy is the derivative of the loss function with respect to y, and dy/dW1 and dy/dW2 are the derivatives of y with respect to W1 and W2, respectively.

The integration part of the backpropagation algorithm uses the gradient to update the model parameters in the direction of the negative gradient, in order to minimize the loss function. The updated model parameters can be calculated as follows:

$$W_1 = W_1 - learning\ rate \cdot \frac{dL}{dW_1}$$

$$W_2 = W_2 - learning\ rate \cdot \frac{dL}{dW_2}$$

where *learning_rate* is a hyperparameter that controls the size of the update.

The Optimization Algorithm:

The optimization algorithm is used to update the network's weights to minimize the loss. Common optimization algorithms include gradient descent, stochastic gradient descent (SGD), and ADAM.

$$w_n = w_o - \alpha \cdot \nabla Loss$$

where w_o and w_n are the old and new weights, respectively, and ∇Loss is the gradient of the loss with respect to the weights.

The Loss Function: Sparse Categorical Cross Entropy

The mathematical formula for the sparse categorical cross-entropy loss function is given by:

$$Loss = -\sum_i (y_t \cdot log(y_p))$$

where y_t is the true label for a particular data sample, y_p is the predicted probability for that label produced by the model, and the summation is performed over all possible labels.

The logarithm term ensures that the loss increases as the predicted probability diverges from the true label.

The Activation Function: SoftMax

The mathematical formula for the SoftMax function is given by:

$$softmax(z) = \frac{\exp(z_i)}{\sum(\exp(z_j))}$$

where z is a vector of arbitrary real-valued inputs and exp is the exponential function. The SoftMax function computes a probability distribution over K possible classes, where each class is represented by a node in the output layer of a neural network. The numerator of the formula computes the exponential of each input, and the denominator computes the sum of these exponentials. The resulting probabilities are then normalized such that they sum to 1.

3.2: Mathematics of the Autoencoder

These equations provide a high-level overview of the mathematics behind designing the architecture of the Auto Encoders for dimensionality reduction.

Reconstruction Loss Function:

The mathematical foundations of autoencoders are based on a reconstruction loss function that measures the difference between

the original input data and its reconstructed output. The most common form of reconstruction loss is the mean squared error (MSE) loss:

$$Loss = 1/n \cdot \sum (x_i - x_i')^2$$

where x_i is the i-th element of the original input data and x'_i is the corresponding reconstructed output.

This equation calculates the average squared difference between the original input and its reconstructed output, which represents the degree of dissimilarity between the two. Minimizing this loss function results in an autoencoder that can effectively reconstruct the input data with minimal loss.

Encoding:

The input data is transformed into a lower-dimensional representation through a series of matrix operations and non-linear transformations. This representation is known as the encoding and is computed using the following equation:

$$h = f(W.x + b)$$

where h is the encoding, x is the input data, W is the weight matrix, b is the bias vector, and f is the non-linear activation function.

Decoding:

The encoding is then transformed back into the original dimension to obtain the reconstructed output, using the following equation:

$$x' = g(W'.h + b')$$

where x' is the reconstructed output, W' is the transposed weight matrix, b' is the bias vector, and g is the non-linear activation function.

B. Datasets and Materials

Dataset:

The data we are using in this project is downloaded from the NASA Exo-Planet Archive[a]. The dataset consists of results from the Kepler Object of Interest (KOI) and TESS Data (Transiting Exoplanet Survey Satellite), collected by the Kepler mission and TESS mission that revealed thousands of planets out of our Solar System. This dataset has already been evaluated whether the detected exoplanet is a *candidate* (a true exoplanet) or *false positive* (not an exoplanet but had been detected as by the satellite). Thereby the previously evaluated results and the major features for evaluation will serve as our dataset in this project to detect the false positive candidate for the future data.

System Requirements:

Processor Configuration: x86 based or ARM based CPU, minimum Dual Core CPU of 2.5GHz and more RAM Configuration: Minimum 4 GB and more Additional Requirement: A GPU of Minimum 4GB can be used for faster training and processing of data

Open Source Python Libraries:

Pandas [b]: It is a python library which is used for analysing, cleaning and manipulation of data. It uses proper and effective data structures to deal with the files easy.

NumPy [c]: It is a python library which is used to perform various operations on arrays and matrices according to the purpose

Seaborn [d]: It is a data visualization library in python which is built over Matplotlib to provide great visuals. It is used to plot the data and gives amplified results

Scikit Learn [e]: It is a robust python library which is used to create statistical modelling which includes

C. Algorithms:

- **Data Pre-processing**

To get better accuracy, the training set should be greater than the testing set so that the model can work on more number of rows. Using Sci-Kit Learn [e], we can easily convert the dataset into training and testing set according to our interest of separation. *Test_size* is the parameter for determining the test set size in scikit learn. *Random_state* is another parameter which allows us to produce the same training and testing set each time we run the code.

- **Dimensionality Reduction Using Autoencoders:**

High dimensional dataset requires more computation and training time. It also requires more space to store the data as it also reduces the performance of the model if the dimensionality of the dataset [20] isn't treated.

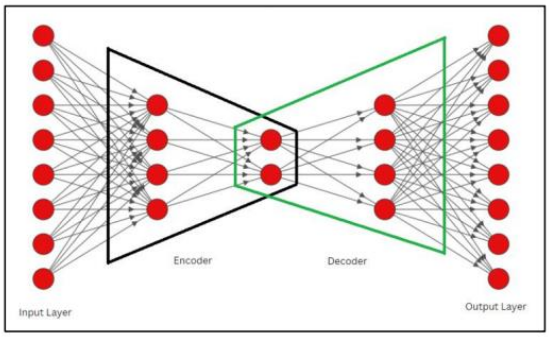


Fig 1: The architecture of the Auto Encoder

To avoid these bottle necks, dimensionality reduction is performed before training of the dataset. Here, we prefer Auto Encoders for dimensionality reduction over PCA (Principal Component analysis) because PCA stores large amount of data in the main memory and it fails if the storage exceeds [19] . This problem is resolved with Autoencoders because it can able to work on smaller batches, so we can avoid the memory limitations. Since the dataset we have used has 42 columns, it is considered as high dimensional dataset [13]. Thereby we will be implementing Autoencoders on this dataset for dimensionality reduction. We use the training and testing set to build the Autoencoder model. We use *fit_transform*() to convert the data into certain datapoints for the model. Encoder object is built, which is also the bottleneck. It has hidden units, that reduces the number of features to the specified units. Then we define the decoder object to reconstruct the compressed input encoder contains. After training the Autoencoder model, we get the data in less number of features or in other words, the number of features that we have

specified as hidden units in the bottleneck. Hence, we got the features scaled down, making Artificial Neural Network to give better and accurate prediction.

- **Artificial Neural Networks for predicting false positivity**

Artificial Neural Networks architecture consists of Input Layer, Hidden Layer and Output as shown in Fig 2.

Here we are using Artificial Neural Networks because we have more than one hidden layer. Giving the optimal number of neurons and activation functions in each hidden layer yields better results. The output layer must contain the number of classes for classifying it whether it is a false positive or a candidate [9].

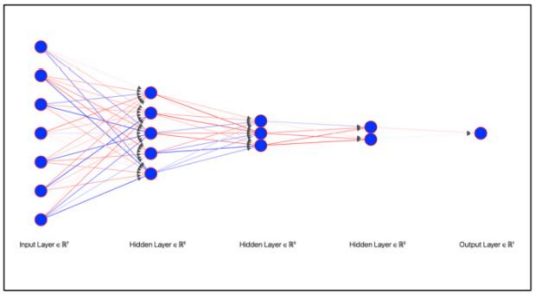


Fig 2: The architecture of the Artificial Neural Network

Activation Layer for hidden layers: ReLU[9]

ReLU[9] stands for the rectified linear unit activation function. We have preferred ReLU[9] for our hidden layers over other activation functions because it does not activate all the neurons at the same time. Only certain neurons get activated at a particular input value range, which makes our ANN model computationally efficient.

Activation function for Output Layer: Softmax[10]

Softmax[9] is ideal for multiclass classification to calculate the probability distribution for a datapoint belonging to each individual class. We have used softmax[10] to show both the probabilities separately even though the result type is binary.

Optimizer used: ADAM[11]

As per the Keras open-source library, Adam[11] Optimizer is defined as a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments. We have preferred this optimizer because Adam[9] is computationally efficient, has less memory requirement and is well suited for large datasets.

Loss Function[29]

SparseCategoricalCrossEntropy computes the loss between the label and predictions. We use this function when we have two or more label classes, as in our dataset.

Metrics Function[29]

Metrics function is used to evaluate the performance of our Artificial Neural Network.

Algorithm for the ArtAe:

import_library tensorflow

import_library sklearn

 # Define the architecture of the autoencoder

Inputs<--Input(shape=(40,))

...

decoded <-- defining the dense layer

autoencoder <--Model(inputs,decoded)

#A model grouping layers into an object with training/inference features.

 # Train the autoencoder

Train(autoencoder(X_train,X_train))

 # Compress input data using the autoencoder

X_compressed<--Predict(autoencoder(X_train))

 # ANN

ann = Sequential()

...

Add(ann(Dense(units=6,activation=Softmax())))

 # Train the neural network on the compressed data

Train(ann(X_compressed,y_train))

 # Make predictions using the trained neural network

preds<--Predict(ann(predict(autoencoder(X_test))))

 # Accuracy

accuracy = accuracy_score(r,y_test)

print(accuracy,"%")

4. Graphical Interpretation

The graphical interpretations of the data offer insight into the model's internal processes, facilitating debugging, optimization, and a deeper understanding of the rationale behind the model's specific predictions.

A. Study of the Orbital Period and Transit Epoch (Kepler Dataset)

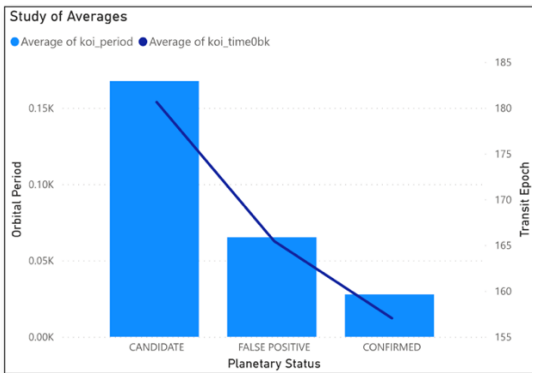


Fig. 3: (a) This plot denotes the average of the values of the orbital period and transit epoch

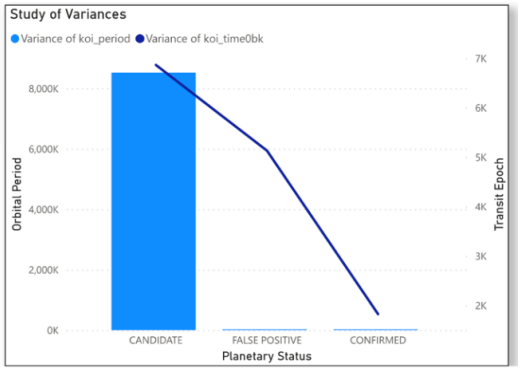


Fig. 3: (b) This denotes variances of the values of the orbital period and transit epoch.

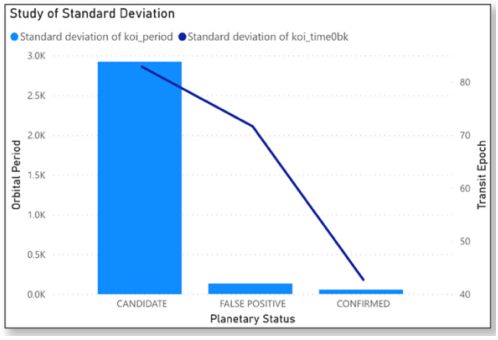


Fig 3: (c) The lower leftmost side denotes the standard deviation of the values of the orbital period and transit epoch

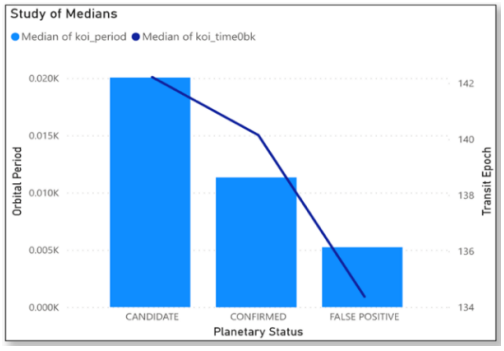


Fig 3: (d) rightmost side denotes medians of the values of the orbital period and transit epoch

The study of averages and variances for orbital periods indicates that confirmed planets tend to have lower average orbital periods compared to false positives. Additionally, the variance in orbital periods is significantly higher for confirmed planets. This suggests that there might be more variation in the orbital periods of confirmed planets, while false positives show a more clustered range of values *(as shown in Fig 3.(a) and (b))*

The analysis of standard deviation and medians reveals that confirmed planets have lower standard deviation in comparison to false positives. Moreover, false positives exhibit lower median values for certain parameters when compared to candidates and confirmed planets. This suggests that confirmed planets have more consistent and less variable data for these parameters, while false positives exhibit greater variation. *(as shown in Fig 3.(c) and (d))*

B. Study of the Transit Duration and Impact parameter (Kepler Dataset)

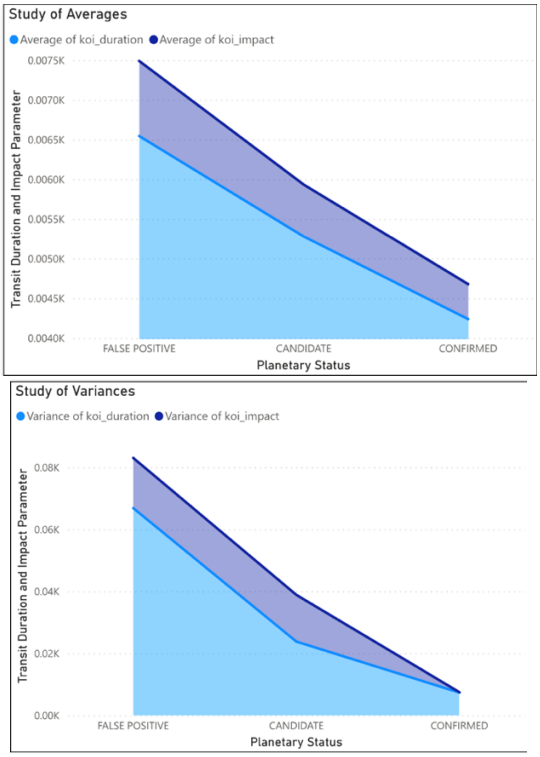
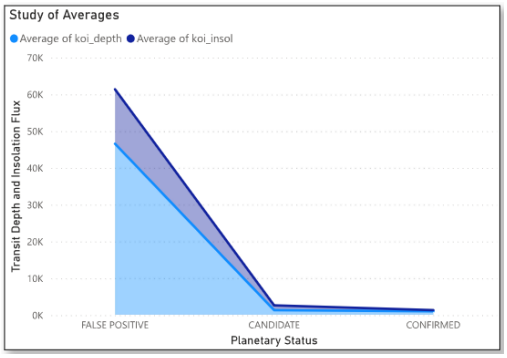


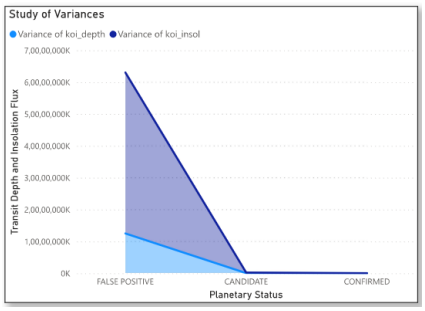
Fig 4: The upperfigure denotes the average of the values of the transit duration and impact parameter and the lower figure denotes variances of the values of the transit duration and impact parameter

In the study of averages for transit duration and impact parameter, confirmed planets show lower average values than false positives. The trend continues in the study of variances, indicating that the data for confirmed planets is more consistent in terms of these parameters. This consistency implies that confirmed planets have more stable and predictable transit durations and impact parameters compared to false positives *(as shown in Fig 3).*

C. **Study of the Insolation and Transit Depth (Kepler Dataset)**



D.



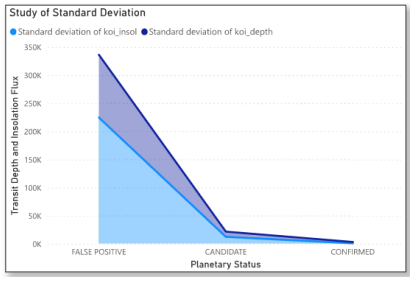
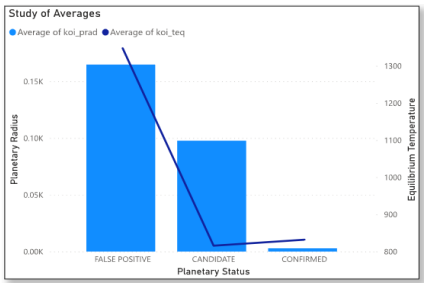


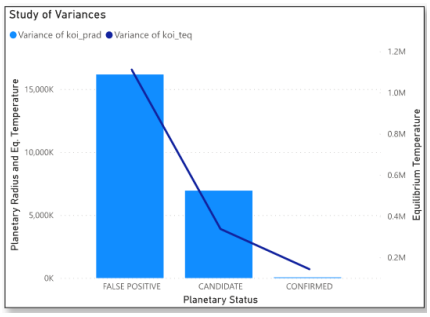
Fig 5: Uppermost figure denotes the average of the values of the Insolation and Transit Depth and middle figure denotes variances of the values of the Insolation and Transit Depth. Lowermost Graph is the standard deviation of the values of the Insolation and Planetary Depth

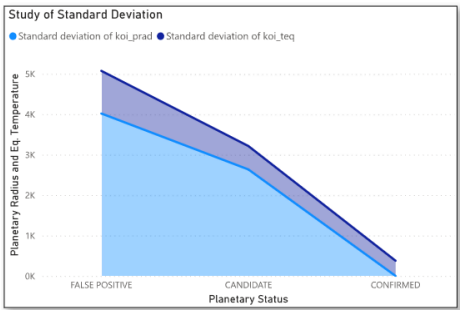
The analysis of averages for planetary depth and insolation values shows that both confirmed and candidate planets have lower mean values than false positives. Notably, confirmed planets exhibit even lower means than candidates. The investigation of variances reveals that false positives have a significantly larger variance, suggesting that their data is less consistent compared to confirmed and candidate planets *(as shown in Fig 5.).*

The comparison of standard deviations implies that candidates have higher standard deviations than confirmed planets in terms of planetary depth and insolation. This could indicate that confirmed planets have more consistent data for these parameters *(as shown in Fig 5).*

E. Study of the Planetary Radius and Equilibrium Temperature (Kepler Dataset)







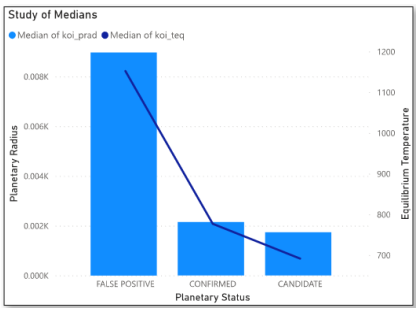


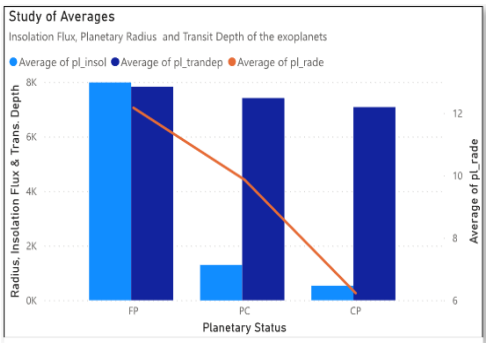
Fig 6: a) The uppermost figure denotes the average of the values of the planetary radius and equilibrium temperature and the second figure denotes variances of the values of the planetary radius and equilibrium temperature.

b) The third figure denotes the standard deviation of the values of the planetary radius and equilibrium temperature and the fourth figure denotes medians of the values of the planetary radius and equilibrium temperature.

The study of averages and variances for planetary radius and equilibrium temperature demonstrates that confirmed planets generally have smaller mean values than false positives and candidates. Furthermore, the variance of confirmed planets is notably lower, indicating that their data is more tightly clustered around the mean. This consistent data suggests that confirmed planets have a more well-defined range of planetary radii and equilibrium temperatures *(as shown in Fig 6.a).*

The analysis of standard deviations for planetary radius and equilibrium temperature confirms that confirmed planets have lower standard deviations compared to false positives and candidates. This suggests that the data for confirmed planets is less spread out, signifying greater consistency. Additionally, in the absence of information about the median plot, it's challenging to make conclusions regarding medians *(as shown in Fig 6.b).*

F. Study of the Isolation Flux, Planetary Radius and Transit Depth of Exoplanets (TESS Dataset)



G.

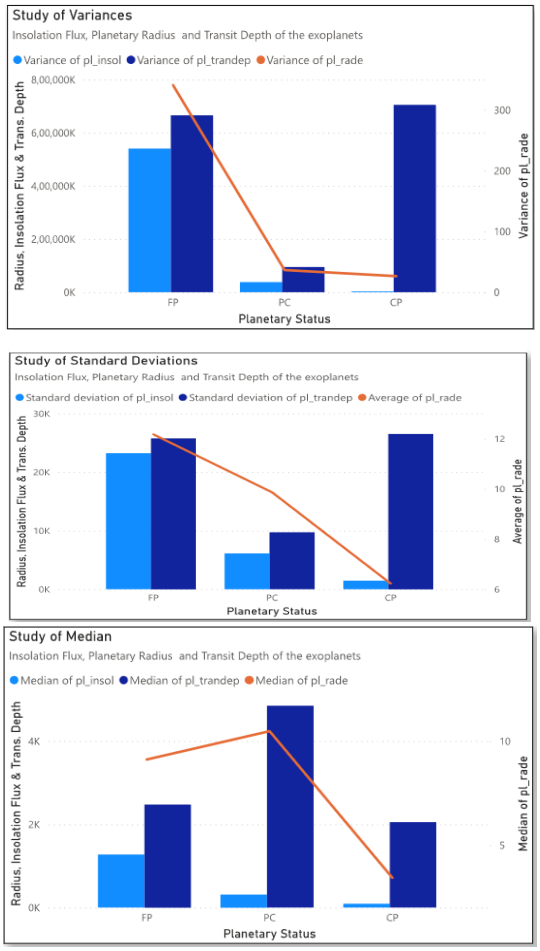


Fig 7: a) The first figure denotes the average of the values of the Isolation Flux, Planetary Radius and Transit Depth of Exoplanets and the second one denotes variances of the values of the Isolation Flux, Planetary Radius and Transit Depth of Exoplanets.

b) The third figure denotes the standard deviation of the values of Isolation Flux, Planetary Radius and Transit Depth of Exoplanets and the last figure denotes medians of the values of the Isolation Flux, Planetary Radius and Transit Depth of Exoplanets.

The analysis of average values for Insolation Flux, Planetary Radius, and Transit Depth reveals distinct characteristics of confirmed planets. Confirmed planets exhibit the lowest mean values in these parameters compared to false positives and planetary candidates. This suggests that confirmed planets tend to have smaller radii, lower insolation levels, and less substantial transit depths, potentially indicating properties conducive to confirmation.

In terms of standard deviation, confirmed planets consistently display the least variation in Insolation Flux and Planetary Radius. However, Transit Depth values show higher standard deviations for both confirmed and false positive planets, indicating a degree of inconsistency in this feature among these groups *(as shown in Fig 7.a)*.

The examination of variances in Insolation Flux, Planetary Radius, and Transit Depth reaffirms the consistency of confirmed planets, as they consistently exhibit the lowest variance in these parameters. Transit Depth variances, on the other hand, remain relatively high for both confirmed and false positive planets, suggesting variability in this feature.

Median values for these parameters reveal that confirmed planets tend to have the lowest median values for Insolation Flux and Planetary Radius. However, Transit Depth median values are notably higher for planetary candidates. False positives and confirmed planets show similar Transit Depth medians compared to planetary candidates *(as shown in Fig 7.b)*.

H. Study of the Planetary Orbital Period and Transit Duration of Exoplanets (TESS Dataset)

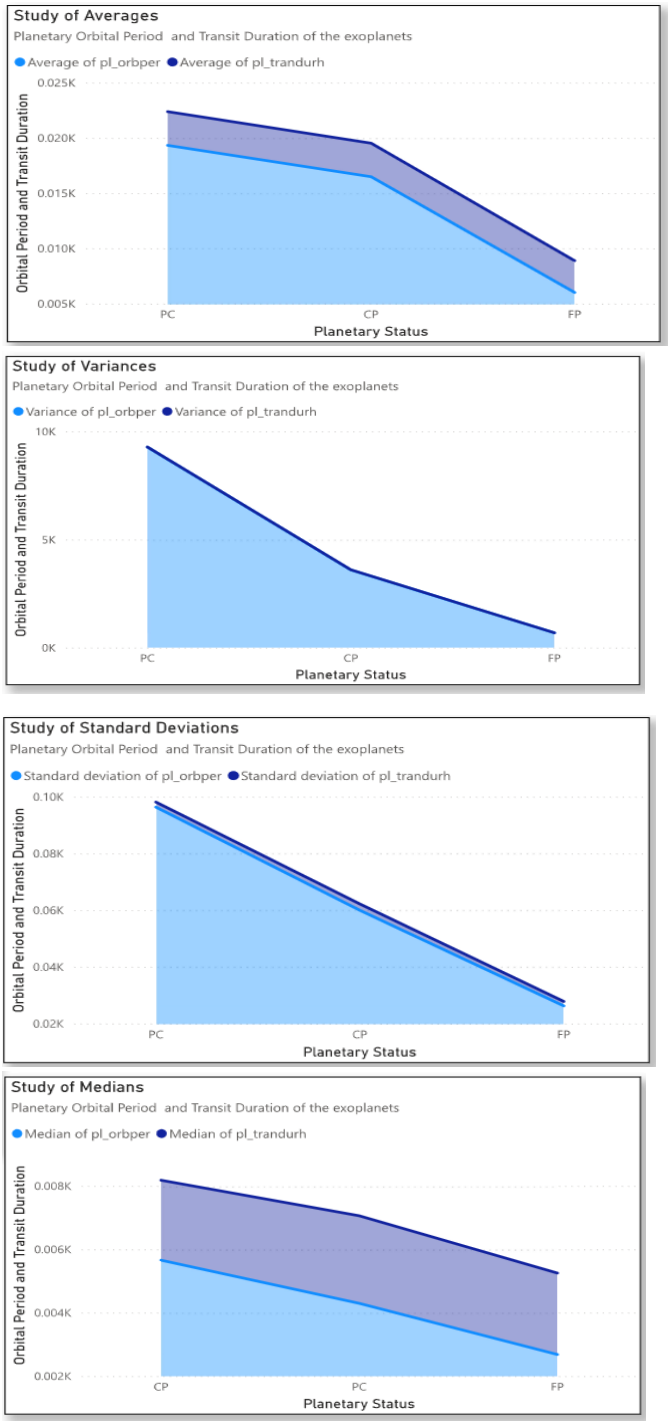


Fig 8: a) The first figure denotes the average of the values of the Planetary Orbital Period and Transit Duration of Exoplanets and second one denotes variances of the values of the Planetary Orbital Period and Transit Duration of Exoplanets.

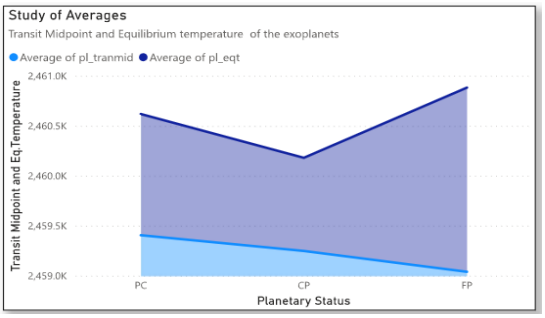
b) The third figure denotes the standard deviation of the values of Planetary Orbital Period and Transit Duration of Exoplanets and

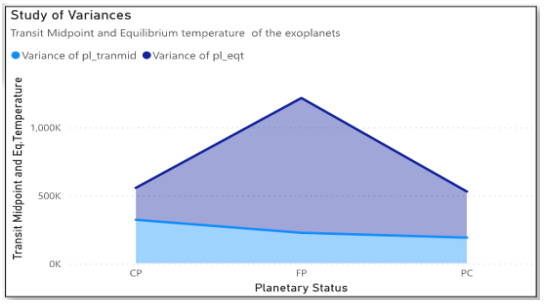
the last figure denotes medians of the values of Planetary Orbital Period and Transit Duration of Exoplanets.

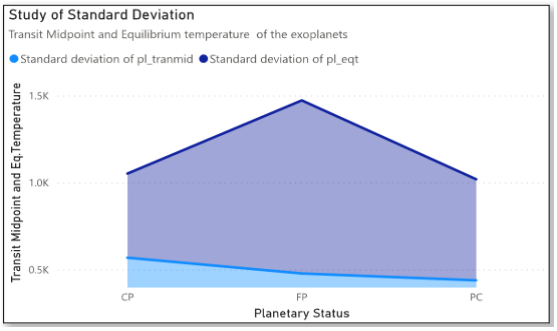
The analysis of average values for Orbital Period and Transit Duration highlights differences between confirmed planets, false positives, and planetary candidates. False positives consistently display the lowest mean values for both Orbital Period and Transit Duration, while planetary candidates exhibit the highest means. This indicates that confirmed planets tend to have longer orbital periods and transit durations compared to false positives. Variance measurements for Orbital Period and Transit Duration align with the trend observed in averages, with false positives consistently having the lowest variances, suggesting more consistent data for these features among false positives *(as shown in Fig 8.a)*.

Standard deviation values for Orbital Period and Transit Duration continue to support the consistency of false positive data, as they consistently exhibit the lowest standard deviations. In contrast, confirmed planets have the highest standard deviations for Transit Duration, indicating greater variability in this property among confirmed planets.Median values for Orbital Period and Transit Duration mirror the trend observed in averages, with false positives consistently showing the lowest median values, while confirmed planets have the highest Transit Duration medians *(as shown in Fig 8.b)*.

I. Study of the Transit Midpoint and Equilibrium Temperature of Exoplanets (TESS Dataset)







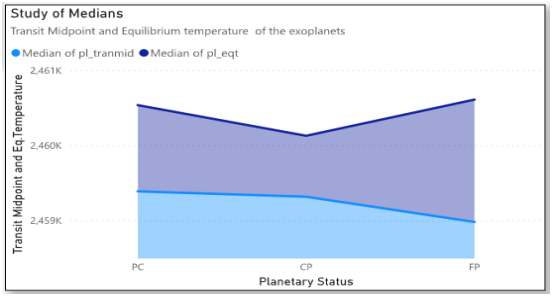


Fig 9: a) The first denotes the average of the values of the Transit Midpoint and Equilibrium Temperature of Exoplanets and side denotes variances of the values of the Transit Midpoint and the second one depicts Equilibrium Temperature of Exoplanets.

b) The third figure denotes the standard deviation of the values of Transit Midpoint and Equilibrium Temperature of Exoplanets and the last figure denotes medians of the values of Transit Midpoint and Equilibrium Temperature of Exoplanets.

The analysis of average values for Transit Midpoint and Equilibrium Temperature (EQT) reveals distinctions between the three classes. False positives have the lowest mean values for Transit Midpoint but the highest mean values for EQT. In contrast, confirmed planets exhibit the least mean EQT values compared to candidates and false positives. These observations suggest differences in the orbital and thermal properties of these groups. Variance measurements indicate that both confirmed, and candidate planets have lower variances for EQT, implying more consistent temperature data in these groups. However, false positives exhibit higher variances for EQT, indicating greater variability in equilibrium temperatures among false positives *(as shown in Fig 9.a)*.

Standard deviation analysis reiterates the consistency of confirmed and candidate planets in terms of EQT, with both groups displaying lower standard deviations compared to false positives. Confirmed planets have the highest standard deviation for Transit Midpoint, suggesting greater variation in this parameter among confirmed planets.Median values further emphasize the differences in EQT between the groups, with false positives displaying the highest median EQT values. However, confirmed planets exhibit the highest median Transit Midpoint values. Candidates and false positives have similar median EQT values *(as shown in Fig 9.b)*.

5. Results

A. Model Results

When applied to the Kepler dataset, it achieved a good accuracy of 93.67% in correctly identifying exoplanets within this dataset. This high accuracy demonstrates that the model has successfully eliminated the false positive result, thereby precisely classifying only the true exoplanet candidates.

B. Performance on TESS Dataset

In the evaluation using the TESS dataset, ArtAe showed remarkable accuracy of 92.10%. This level of accuracy illustrates the model's adaptability across distinct datasets and its capacity to provide accurate exoplanet assessments.

C. Dimensionality Reduction using AutoEncoders

The study also revealed that the time required to train the Artificial Neural Network (ANN) model with dimensionality reduction through Auto Encoders was 74.86 seconds, while the time to train the ANN model without dimensionality reduction was 73.98 seconds, resulting in a difference of 0.88 seconds for a dataset of 7803 records. The use of dimensionality reduction helps reduce overfitting while saving time.

Additionally, the loss during the training of the data was reduced from 83.5277 to 0.7753 after 100 epochs, which shows the efficiency of the integration of the two models. This marked improvement in training performance indicates the effectiveness of combining the ANN and Auto Encoder models for data processing.

The Auto Encoder was successful in reducing the dimensionality of the data from 7803 rows and 42 features (41 features and 1 target column) to 7803 rows and 31 features (30 features and 1 target column).

This was noted as a part of the study only for Kepler dataset.

This reduction in the number of features helps the model focus on the most important characteristics of the data, eliminates noise, and results in improved performance and reduced overfitting.

The use of dimensionality reduction techniques helps ensure that the ANN model effectively learns the underlying patterns and relationships in the data, leading to improved predictions and better overall results.

6. Discussions and Applications:

Discussions:

While the use of dimensionality reduction proves to be beneficial in reducing overfitting and time consumption, there is a trade-off between accuracy and feature reduction. Proper optimization of the model can lead to a reduction in both time and loss, ultimately resulting in higher accuracy. This can be achieved through hardware configurations such as using a GPU or TPU (if TensorFlow package is used) and having an adequate amount of RAM.

It is important to note that this model can also be trained with other exoplanet datasets such as K2 (Kepler Second Light), COROT (Convection, Rotation and Planetary Transits), etc.

Artificial Neural Networks (ANNs) and Autoencoders have a number of advantages that make them attractive for various applications.

One of the main advantages of ANNs is their ability to learn from data and make predictions based on that learning [12]. This ability makes them suitable for a wide range of applications, from image recognition and natural language processing to stock market predictions and disease diagnosis.

Another advantage of ANNs is their ability to handle non-linear relationships between inputs and outputs, which makes them more versatile than traditional machine learning models that are based on linear relationships [9].

Autoencoders, on the other hand, have the advantage of being able to reduce the dimensionality of data while preserving important features [13]. This reduction in dimensionality can lead to improved performance in many applications, including anomaly detection, recommendation systems, and data compression.

However, there are limitations to this approach as reducing features can result in overfitting, while retaining a high number of features can increase processing time. Thus, finding the optimal balance between feature reduction and accuracy is crucial in order to effectively train the model.

Applications:

The ArtAe model has potential applications in handling satellite data by space organizations. Specifically, its ability to effectively differentiate genuine exoplanet candidates from false positives using only transit properties could be useful for identifying and characterizing exoplanets discovered by space-based telescopes.

In addition, the ArtAe model's ability to process large datasets efficiently could be valuable in handling the vast amounts of data generated by space-based instruments[21]. This could include data from Earth observation satellites or other scientific instruments, where the efficient handling and analysis of large datasets is critical[22].

The ArtAe model could also have potential applications in spacecraft operations. For example, during the operation of spacecraft, various sensors and instruments generate large amounts of data [22][27] that must be monitored and analysed to ensure the proper functioning of the spacecraft [25].

This model can process data in real-time could be particularly valuable for spacecraft operations, where quick decisions and responses may be necessary in critical situations.

7. Acknowledgement

We, Nidhi Shakhapur and Ravin D, the authors of this paper, would like to express our heartfelt gratitude to Dr. Sathya K for her invaluable mentorship and unwavering support throughout the research process. Her guidance and expertise have been instrumental in shaping this work. We also extend our sincere thanks to VIT University for providing us with the necessary resources and environment to conduct our research. Their support has been instrumental in our academic journey. Furthermore, we would like to acknowledge the contributions of SEDS VIT student chapter at VIT and its dedicated chair, Chetan Shiraguppi. Their collaboration and assistance have enriched our research experience and added depth to our work. Our acknowledgement goes out to all those who have contributed to this endeavour in various ways. Your support has been pivotal in the successful completion of this research paper.

References

[1] MacDonald, R. J. (2023, January 13). POSEIDON: A Multidimensional Atmospheric Retrieval Code for Exoplanet Spectra. Journal of Open-Source Software, 8(81), 4873. https://doi.org/10.21105/joss.04873

[2] Dimitrov, N., & Natarajan, A. (2019, May 1). From SCADA to lifetime assessment and performance optimization: how to use models and machine learning to extract useful insights from limited data. Journal of Physics: Conference Series, 1222(1), 012032. https://doi.org/10.1088/1742-6596/1222/1/012032

[3] Martin, S. R., Szwaykowski, P., & Loya, F. M. (2005, October). TPF-I Planet Detection Testbed: Progress in Testing Exo-planet Signal Detection. Proceedings of the International Astronomical Union, 1(C200), 279–284. https://doi.org/10.1017/s1743921306009458

[4] Lin, Wu, Fu, Wang, Zhang, & Kong. (2019, October 28). Dual-NMS: A Method for Autonomously Removing False Detection Boxes from Aerial Image Object Detection Results. Sensors, 19(21), 4691. https://doi.org/10.3390/s19214691

[5] Baluev, R. (2018, October). PlanetPack3: A radial-velocity and transit analysis tool for exoplanets. Astronomy and Computing, 25, 221–229. https://doi.org/10.1016/j.ascom.2018.10.005

[7] Espinoza, N. (2018, November 12). Efficient Joint Sampling of Impact Parameters and Transit Depths in Transiting Exoplanet Light Curves. Research Notes of the AAS, 2(4), 209. https://doi.org/10.3847/2515-5172/aaef38

[8] Alpaydin, E. (2010). Introduction to machine learning (2nd ed.). Cambridge, MA: MIT Press.

[9] Bengio, Y., Goodfellow, I. J., & Courville, A. (2015). Deep learning. Cambridge, MA: MIT Press.

[10] Brownlee, J. (2020). Neural networks for computer vision: A gentle introduction. Machine Learning Mastery. https://machinelearningmastery.com/neural-networks-for-computer-vision/

[11] Haykin, S. (2009). Neural networks and learning machines (3rd ed.). Upper Saddle River, NJ: Pearson Education.

[12] Hsu, S., Zhang, H., & Xing, E. P. (2019). Neural network models for joint dimensionality reduction and clustering. Proceedings of the AAAI Conference on Artificial Intelligence, 33, 517-524.

[13] Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B. (2015). Adversarial autoencoders. arXiv preprint arXiv:1511.05644.

[14] Zhou, Y., Wen, D., & Fan, Y. (2017). Auto-encoder based anomaly detection for temporal data. IEEE Transactions on Cybernetics, 47(12), 4109-4121.

[15] Shallue, C. J., & 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), 94.

[16] Malik, A., Moster, B. P., Obermeier, C., Exoplanet detection using machine learning, Monthly Notices of the Royal Astronomical Society, 513(4), 5505-5516. https://doi.org/10.1093/mnras/stab3692.

[17] Liu, L., & Deng, J. (2018, April 29). Dynamic Deep Neural Networks: Optimizing Accuracy-Efficiency Trade-Offs by Selective Execution. Proceedings of the AAAI Conference on Artificial Intelligence, 32(1). https://doi.org/10.1609/aaai.v32i1.11630.

[18] Sturrock, G. C., Manry, B., & Rafiqi, S. (2019). Machine Learning Pipeline for Exoplanet Classification. SMU Data Science Review, 2(1), 9.

[19] Catanzarite, J. H. (2015). Autovetter Planet Candidate Catalog for Q1-Q17 Data Release 24, (K SCI-19090-001). NASA Ames Research Center.

[20] Ansdell, M., Ioannou, Y., Osborn, H. P., Sasdelli, M., Smith, JC., Caldwell, D., et al. (2018). Scientific Domain Knowledge Improves Exoplanet Transit Classification with Deep Learning. The Astrophysical Journal, 869(1), L7.

[21] Yu, L., Vanderburg, A., Huang, C., Shallue, CJ., Crossfield, IJM., Gaudi, BS., et al. (2019). Identifying Exoplanets with Deep Learning. III. Automated Triage and Vetting of TESS Candidates. The Astronomical Journal, 158(1), 25.

[22] Chintarungruangchai, P., & Jiang, IG. (2019). Detecting Exoplanet Transits through Machine-learning Techniques with Convolutional Neural Networks. Publications of the Astronomical Society of the Pacific, 131(1000), 064502.

[23] Fiscale, S., et al. (2023). Identifying Exoplanets in TESS Data by Deep Learning. In: Esposito, A., Faundez-Zanuy, M., Morabito, F.C., Pasero, E. (eds) Applications of Artificial Intelligence and Neural Systems to Data Science. Smart Innovation, Systems and Technologies, vol 360. Springer, Singapore.

[24] Valizadegan, H., et al. (2022). ExoMiner: A Highly Accurate and Explainable Deep Learning Classifier That Validates 301 New Exoplanets. The Astrophysical Journal, 926(2), 120.

[25] Gupta, T. K., & Kumar, C. (2015, June 25). Deep Autoencoders for Non-Linear Dimensionality Reduction. Journal of Bioinformatics and Intelligent Control.

[26] Liang Yu, Andrew Vanderburg, Chelsea Huang, Christopher J. Shallue, Ian J. M. Crossfield, B. Scott Gaudi, Tansu Daylan, Anne Dattilo, David J. Armstrong, George R. Ricker 2019, "Identifying Exoplanets with Deep Learning. III. Automated Triage and Vetting of TESS Candidates," The Astronomical Journal, Volume 158, Number 1.

[27] Anne Dattilo, Andrew Vanderburg, Christopher J. Shallue, Andrew W. Mayo, Perry Berlind, Allyson Bieryla, Michael L. Calkins, Gilbert A. Esquerdo, Mark E. Everett, Steve B. Howell 2019, "Identifying Exoplanets with Deep Learning. II. Two New Super-Earths Uncovered by a Neural Network in K2 Data," The Astronomical Journal, Volume 157, Number 5.

[28] Tey, E., Moldovan, D., Kunimoto, M., Huang, C. X., Shporer, A., Daylan, T., Muthukrishna, D., Vanderburg, A., Dattilo, A., Ricker, G. R., & Seager, S. 2023, "Identifying Exoplanets with Deep Learning. V. Improved Light Curve Classification for TESS Full Frame Image Observations." Retrieved from https://doi.org/10.48550/arXiv.2301.01371

[29] Chatterjee, Supratik & Keprate, Arvind. (2021). Predicting Remaining Fatigue Life of Topside Piping Using Deep Learning. 10.1109/ICAPAI49758.2021.9462055.

[a] NASA Exoplanet Archive: https://exoplanetarchive.ipac.caltech.edu/

[b] Pandas: https://pandas.pydata.org/docs/

[c] NumPy: https://numpy.org/doc/

[d] Seaborn: https://seaborn.pydata.org/

[e] Scikit-Learn: https://scikit-learn.org/stable/index.html