National College of Ireland

Predicting Exoplanet Habitability Using Machine Learning

MSc Research Project
MSc in Data Analytics

Debayan Biswas
Student ID: x22242821

School of Computing
National College of Ireland

Supervisor: Dr. Bharat Agarwal

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Debayan Biswas
Student ID:	x22242821
Programme:	MSc in Data Analytics
Year:	2024
Module:	MSc Research Project
Supervisor:	Dr. Bharat Agarwal
Submission Due Date:	12/08/2024
Project Title:	Predicting Exoplanet Habitability Using Machine Learning
Word Count:	6398
Page Count:	20

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:	Debayan Biswas
Date:	11th August 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):	

Predicting Exoplanet Habitability Using Machine Learning

Debayan Biswas
x22242821

Abstract

The study of exoplanets is a vital area for understanding the presence of life beyond the solar system. Despite having significant scientific achievements and findings in this field, there remains a significant gap in the development of methodologies to accurately predict and classify the habitability of exoplanets. The motivation of this study comes from the limitations of the present methodologies due to the presence of imbalance class and data scarcity in the data recorded due to the vastness of space. This project aims to overcome this problem by integrating data from the Transiting Exoplanet Survey Satellite (TESS) and the Planetary Habitability Laboratory (PHL) sources followed by the application of an extended Conditional Generative Adversarial Networks (cGANs) algorithm. This extended algorithm will lay forward the creation of a robust model to handle complex astronomical data by using a custom classifier XGBoost. The GANs will aid in generating real-life synthetic data thereby enhancing the existing dataset. The core findings of this research is to show the effectiveness of the cGAN in generating synthetic data and using custom classifier in predicting potentially habitable exoplanets. The result of the project show the the prediction accuracy achieved is at 96% that determine the accuracy and efficiency of the model. The contribution of this project can open a new chapter in this astronomical research domain by enhancing exoplanet research and thereby making way for future voyages toward our future destination.

1 Introduction

The journey to know the unknown and what lies beyond our solar system has been an important subject for mankind. The search for a new habitable planet has already begun to support the ever-increasing demand for human settlement. To date, more than 3800 exoplanetary systems have been discovered Rojas-Ayala (2023) out of which the majority are invisible to the naked eye due to their distance from us. With the advancement in technology, the detection of distant world is now possible with the aid of scientific telescopes and space exploration techniques. The space data collected from Transiting Exoplanet Survey Satellite (TESS) have provided with an array of data on exoplanet transits while data from Planetary Habitability Laboratory (PHL) have complied extensive dataset on exoplanet characteristics. This study combines the data from these satellites and aims to implement advanced machine learning technique of conditional Generative Adversarial Networks (cGANs) and integrating it with gradient boosting algorithm XGBoost to effectively predict the habitability of exoplanets.

1.1 Motivation

The vastness of space has put certain limitations to accurately classify the habitable exoplanets. The present means of gathering data is limited to ground and space telescopes only. The collection of data from distant object brings forward a lot of data scarcity and class imbalance in the recorded data. Also, the interpretation of these gathered data does not always foolproof the habitability of exoplanets. Hence comes a need to integrate a more complex and advanced measure in dealing with the exoplanet prediction. This motivation paved the way to combine dataset from different sources to enrich the dataset and implement an extended cGAN algorithm with custom classifier XGBoost to solve the prevalent issues.

1.2 Research question

The research question is as follows:

1. How can the integration of TESS and PHL datasets improve the efficiency of predicting potential habitable exoplanets?

2. How can an extended cGANs algorithm be utilized to address class imbalance and improve the classification of potential habitable exoplanets?

The solution to this research question will address the problems of class imbalance by utilizing cGAN. By incorporating a gradient-boosting classifier XGBoost, and applying hyperparameter tuning, this approach is expected to enhance the prediction accuracy.

1.3 Hypothesis

The research hypothesis states that the application of cGAN to handle class imbalance by generating synthetic data similar to the actual data when combined with machine learning technique like XGboost can significantly enhance and improve the accuracy of exoplanet habitability predictions.
The null hypothesis states that the application of cGAN to generate synthetic data when combined with machine learning technique like XGBoost does not improve the accuracy of exoplanet habitability predictions when compared to the state of the art Jakka (2023) which is having 92% to 95% accuracy.

1.4 Report outline

The research report is divided into several sections and sub-sections to enhance the flow and readability of the research. A brief introduction of the research motivation, challenges, research question and hypothesis is presented in Section 1. Section 2 presents the findings of the existing research work and the current state of the art. Section 3 discusses about the flow of the project starting from data collection to final results. Section 4 discusses the pseudo code of this research project. The algorithm implementation, and tools and hardware required for code implementation is discussed in Section 5. The results of exoplanet habitability prediction is presented in the Section 6. Section 7 discusses research findings and elaborates the future scope for this project. Finally, References consist of all cited literature papers and sources.

2 Related Work

This section gives a detailed explanation of current research on exoplanets in a logical order that can provide a direction for further research in the future.

2.1 Finding exoplanets

Detecting an exoplanet is the first step toward assessing its potential habitability. There are several detection techniques present including primarily astronomical observation and planet characterization. Dai et al. (2021) discussed about methods like Radial Velocity (RV), Transit Photometry, Astrometry, and Microlensing. Newman et al. (2023) discussed about generating simulations for exoplanets using RV Surveys. RV survey depends on the frequency of Earth-like exoplanets around their host stars. The surveys are used to identify the potential exoplanets that are hosting their stars by measuring the reflex velocity of the stars. This paper mentioned that a properly designed survey could achieve the necessary sensitivity to detect Earth-like exoplanets with similar mass. Although the observational resource requirement for achieving this is high and requires proper infrastructure. Qiao-Yang et al. (2023) discussed various detection methods such as RV, Transit, Astrometry, and Direct Imaging. The paper highlighted that RV could detect about 2% of Earth-like exoplanets around M stars with high precision, though it's limited by current technology. The Transit method is ideal for detecting Earth-like planets, providing detailed orbital and size information, but requires the transiting orbit to align with the observer. Astrometry offers high precision for planets around G stars, yet is sensitive to stellar noise and requires high-precision measurements. Direct Imaging is effective for exoplanets around bigger stars with significant planet-star separation. Being image data, it offers insight into exoplanet composition, but challenges remain in achieving the necessary resolution and contrast due to noise.

Singh and Singh (2023) analysed the Transit method and RV approach for the exoplanet detection method. The transit method detected dips in the star's brightness while it transits its stellar host and has successfully discovered various exoplanets. This method is limited to low-brightness dips whose orbital periods are long. The observing perspective is important for proper detection of exoplanets. The RV method measures stellar motion by the gravitational pull of orbiting planets thereby predicting the mass and orbital parameters of exoplanets. This process requires the study of the Doppler effect and Redshift on the observed data. Precise measurement is a challenge as RV cannot directly estimate the planet's orbit and can generate false positive signals that simulate the presence of a planet and is unsuitable for lower-mass planets and those orbiting distant host stars. Prasad et al. (2023) focused on the transit method to detect exoplanets by analysing the time series data of light curves. This method identified the brightness dips that occurs when an exoplanet passes in front of their star. These dips called flux can estimate the mass and radius of space objects thereby determining exoplanets. The transit data generated high Sound-to-noise ratio (SNR) ensuring that there is minimal loss of information in the light curve data, which reduced the chance of false negatives. This method is not limited to detecting only brighter planets and will lead the path in detecting further exoplanets with increased precision in calculating mass, radius, orbital velocity, and other parameters.

Although all the methods are effective in detecting exoplanets, the transit method can better predict exoplanets due to high SNR, precise data of the mass, radius, and other parameters, and is not limited to detecting larger planets. The TESS and PHL datasets are generated using the transit method.

2.2 Class imbalance handling

The class imbalance present in data needs to be handled beforehand so that the data can be useful for better classification. The study by Yi et al. (2021) discussed the classical Over-sampling techniques of Synthetic minority over-sampling technique (SMOTE) and a proposed Minority clustering SMOTE (MC-SMOTE) applied for a wind turbine blade icing fault detection. SMOTE addressed issues of minority class by generating synthetic samples through linear interpolation between neighboring minority samples. The distribution of the minority class is not often uniform due to the interpolation leading to subpar performance. However, the proposed approach handled this issue by clustering minority classes first into several samples. The synthetic data was generated using linear interpolation between adjacent clusters rather than neighbor minority classes. This ensured a wider range of minority classes with a more even class distribution. However, the proposed MC-SMOTE suffered in effectiveness as it only showed better results for uneven class distribution. Also, it is susceptible to uncertainties like noise and outliers. A more robust model is necessary to handle these uncertainties. Khoda et al. (2021) discussed about addressing data imbalance issues in malware detection for edge devices in IoT networks. The proposed approach of Fuzzy set theory and Dynamic loss function with class weighting addressed the information loss, over-fitting, and invalid samples that arise from class imbalance in traditional methods. The proposed technique improved the synthetic sample quality thereby enhancing model training. Both method focused on processing uncertainty and offered higher priority or weights to the minority class. The proposed approach showed 9% improvement in the F1 score in malware detection. However, the fuzzy-based approach has limitations due to sensitivity towards user-chosen settings. The paper mentioned about introduction of adversarial retraining techniques that may further enhance stability and effectiveness.

Li et al. (2018) implemented an adversarial network in a novel way called Text-to-text GAN (TT-GAN) in Natural Language Processing (NLP) that can generate realistic text, summaries, and paraphrases. The model generated realistic texts that were similar to the content of the source. The generative model successfully generated paraphrases and semantic summaries showing the capabilities of GAN. However, the model faced challenges related to the generation of discrete text due to the differentiability of GANs. The study mentioned about scaling the capabilities of the GAN model and its further development in this domain as a future scope. Douzas and Bacao (2018) introduced an extended version of GAN called conditional Generative Adversarial Networks (cGAN) as an oversampling method. cGAN unlike SMOTE, approximates the data distribution of minority class. The quality of the synthetically generated data is of higher quality and generated realistic data. The author evaluated the performance of cGAN across 71 datasets with class imbalance and cGAN outperformed other methods across different classifiers like Decision Tree, Logistic regression (LR), Gradient boosting models, etc. cGAN is efficient in handling complex structures, patterns and once trained generates new minority class samples. The generator takes noise and minority class as inputs and

the discriminator generates synthetic data out of it. However, adequate model training is necessary to generate evenly distributed samples that lead to optimal results. The study focused on binary classification but states the potential of cGAN in handling multi-class imbalances. Another cGAN model utilized by Yang (2020) is applied to a flight engine vibration dataset. The cGAN can be conditional as per problem requirement. The author applied a combined approach along with a Support vector machine (SVM) to understand the engine vibration data. The cGAN solved the problem of insufficient realistic flight data by generating new data with the collaborative effect of generator and discriminator. The evaluation depicted that these changes increase the F1 score by 25% along with precision and recall to gain 15.6% and 80% respectively. These collaborations are far better than previous papers.

In summary, while SMOTE and Class Weighting have certain limitations, cGAN approach overcomes these limitations by generating high-quality realistic minority class samples. cGAN being more complex and advanced, preserves the features while generating synthetic samples. The addition of a classifier to cGAN significantly improved the model's performance and hence selecting a suitable classifier is necessary to achieve better results.

2.3 Machine Learning in Exoplanet Research

Bahel and Gaikwad (2022) talked about the use of light-intensity time series data collected from NASA's Kepler mission and using machine learning techniques for exoplanet detection. The study applied a decision tree, LR, and k-Nearest Neighbor (KNN) on rebalanced data using SMOTE. The study showed that KNN performed the best and attained an Accuracy of 98.20% with an F1 score of 98%. KNN also performed well with unbalanced original data attaining high accuracy of 99.1% using only 1% of the data. Decision tree struggled with balanced data where its accuracy and F1 were lower than the unbalanced dataset. LR performed the worst with very low accuracy. KNN showed effectiveness for limited data. The study by Vishwarupe et al. (2022) focused on similar Kepler data to be used for machine learning. The effectiveness of the various machine learning algorithms are compared. Here, Random Forest (RF) classifier was the most efficient with F1 score of 96.2% for handling complex multi-variate data with the presence of minimum noise. The decision tree had a similar performance to the previous paper with F1 score of 93.7% but now is susceptible to noise issues. KNN and LR were unsuitable as they suffered from overfitting and outliers in the data. The study highlighted the need for a more advanced approach for generating better-performing models.

Bhamare et al. (2021) used the Kepler data extracted from Kepler cumulative object of interest (KCOI). Feature selection and preprocessing reduced the feature count drastically to necessary features. Support Vector Machine (SVM), RF and Adaboost was implemented for classification. RF performed well in general compared to previous study with F1 score at 98%. SVM achieved a fair score at 97.72% and these results were due to the carefully selected features. The gradient boosting algorithm Adaboost performed slightly better than RF with F1 at 98.03%. RF and Adaboost were equally capable in predicting exoplanets. Sharma et al. (2023) summarised a study comparing various machine learning algorithms for predicting and classifying space objects like quasars, galaxies and stars. The transit data after preprocessing was passed through Principal Component

Analysis (PCA) which is dimensional reduction technique that preserves variability. The Multiclass LR and the Naïve Bias model being simple models performed well with F1 score at 95.33% and 96.6% on average for the 3 classes. Complex model like Decision Tree performed better than previous study with F1 score at 97.3%. The highest performance was achieved by the gradient boosting model XGBoost with average F1 at 99% for combined classes and 100% accuracy for confirmed class 2.

XGBoost's high accuracy and efficiency make it suitable for real-time data analysis. Both gradient boosting algorithms Adaboost and XGBoost are the best performing and showed their robustness in predicting exoplanets. Extending cGAN with XGBoost classifier will make the model more efficient and effective in predicting and classifying the exoplanet habitability.

2.4 Combined Approach for Prediction

The paper Chen et al. (2024) compared the result of cGAN with or without a classifier in the GAN framework. The study revealed that Energy-based Conditional Generative Networks (ECGAN) that combines a classifier showed better results than cGANs without classifiers. The classifier improved the accuracy of the cGAN leading to better performance on challenging datasets. The downside is that they require more computational resources than normal cGAN on large datasets. The study by Ghaleb et al. (2023) proposed a combination of ensemble learning and a Gan-based Ensemble Synthesized Minority Oversampling Technique (ESMOTE-GAN) used for fraud detection. This approach addressed the presence of high-class imbalance by generating datasets with less noise. The model is extended with a RF classifier that achieved improved performance in detecting fraud with an increased detection and false alarm rate. However, overfitting may arise that needs to be properly treated.

The GAN when extended with a classifier performers better than without classifiers. Thus the proposed cGAN model extended with classifiers XGboost will be robust enough to predict the habitability potential of exoplanets.

3 Methodology

This section provides detailed discussion of the research methodology. The steps from initial data collection to model training is mentioned here in a systematic scientific way, ensuring the validity of the findings. The methodology flow can be seen in Figure 1.

3.1 DATA COLLECTION

The two datasets used for this project are the Transiting Exoplanet Survey Satellite (TESS) NASA Exoplanet Archive (2024) and Planetary Habitability Laboratory (PHL) Planetary Habitability Laboratory (2024) which are sourced from the NASA Exoplanet Archive and the University of Puerto Rico PHL website, respectively. Both the datasets consist of necessary features for the habitability prediction purpose. However, TESS only has features present in the dataset whereas PHL has both features and targets necessary for prediction. Hence, the dataset if combined will enrich the dataset forming a more informative dataset having both features and targets. There are several common

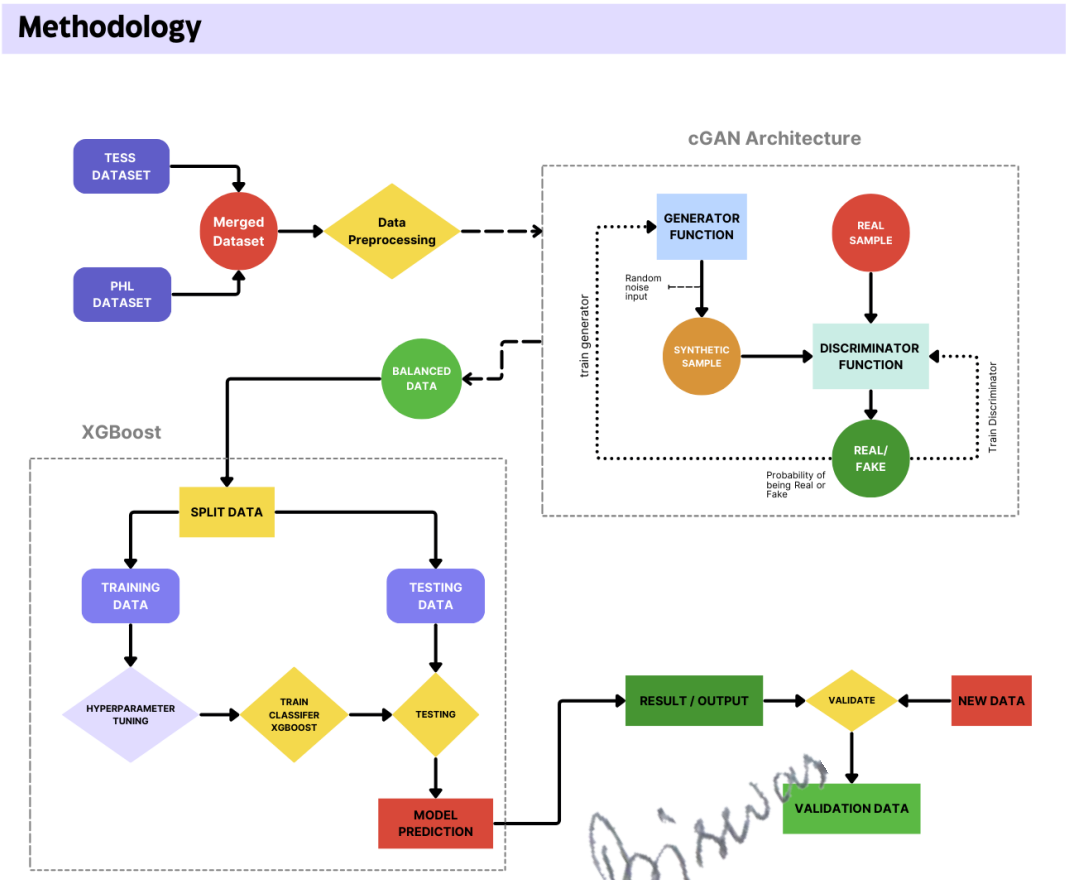

Figure 1: Project methodology

columns present in both the datasets that will be inspected while merging. The process of combining the datasets is mentioned in the next subsection.

3.2 DATA INTEGRATION

The two datasets in use share several common columns and also include additional columns with different sets of information. This is because each dataset being derived from different space telescope, resulting in a significant amount of data scarcity. Combining the datasets will create a more enriched dataset, making it more impactful for habitability predictions. Both datasets contain several redundant features that do not have significant contribution for predictions and are therefore omitted. The TESS dataset consists of only features with no target variable. Integrating TESS and PHL brings the target variable in the dataset which will be used for prediction. The columns defining the planet's mass, radius, orbital period, and eccentricity, as well as the stellar object, stellar radius, mass, metallicity, distance, degree, earth similarity index, and surface temperature are crucial for habitability prediction and are the only factors considered. The most vital factor of habitability which is the target variable, classifies the planets in three categories: 0 for not habitable, 1 for potentially habitable, and 2 for confirmed habitable planets. Class 0 is the majority class consisting of 5331 elements whereas the minority classes 1 and 2 have 22 and 39 elements respectively. The class imbalance in this dataset is significant, with class 0 comprising 98.87% of the data, while the minority classes

occupy the remaining portion. This class imbalance is carefully handled, as discussed in subsection 3.5. After data integration, the merged dataset was checked to ensure accuracy and the absence of discrepancies. However, the merged dataset contains a significant number of missing values checked using msno matrix, as seen in Figure 2, which need to be processed.

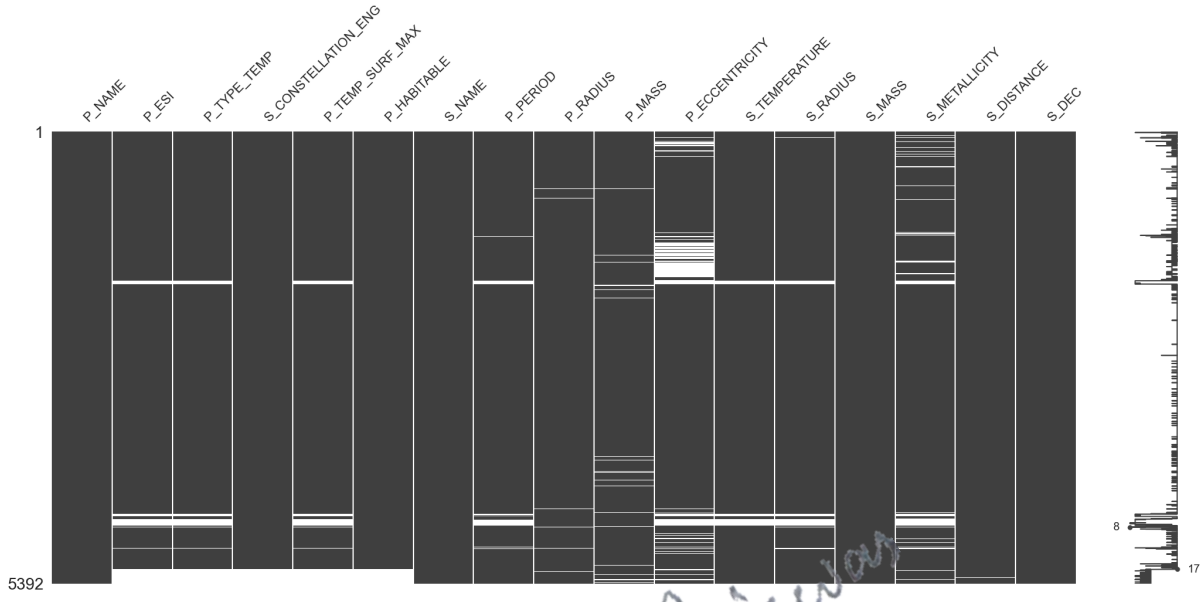

Figure 2: Matrix showing missing values

3.3 DATA PREPROCESSING

Data preprocessing is a vital step in preparing the dataset for imbalance class handling and model training. The merged dataset has missing information which needs to be addressed. The dataset is split into features and target label based on the presence of the target variable P_HABITABLE. The categorical and numerical columns are identified from the features and preprocessing pipelines for both types of columns are created. The numerical pipeline uses a KNN Imputer to fill in missing values by averaging the values from the nearest rows with similar data. The categorical pipeline uses a Simple Imputer approach to fill in the missing values with the most frequent category. Column Transformer is used to effectively apply both the preprocessing techniques to ensure each column is imputed properly. After preprocessing of the numerical and categorical features, the features and target label are merged together. The dataset still has 172 missing rows in the target label which needs to be addressed. The dataset is now split into two parts, one containing rows where P_HABITABLE value is present (known) and the other with P_HABITABLE value missing (unknown). These split dataset will aid in finding the missing target label using an RF classifier. The categorical features are encoded using One-hot encoding to transform them into binary variables needed by the classifier for predicting. The model is trained on the on the known data to predict the missing data which fills the missing gaps in the original dataset. The classifier predicts the missing values based on data patterns and relationship.

3.4 DATA VISUALIZATION

The relationship between the features is explored in this section. The two datasets after merging and preprocessing shows a class distribution which is depicted in Figure 3.

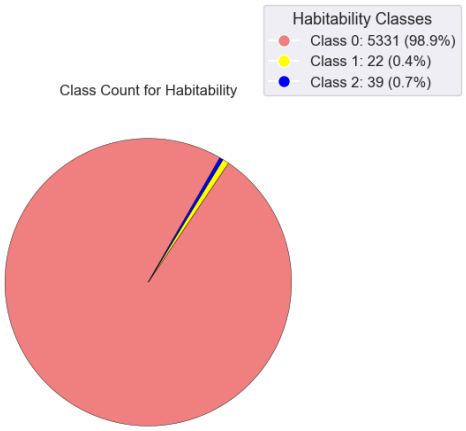

Figure 3: Class Distribution after merging and preprocessing

The plot for the features stellar temperature and mass in relation to habitability depicts the effect of the host star's features on habitability. Figure 4 depicts a linear spread suggesting that the more the stellar mass, the greater the temperature of the star. This suggests that the brighter dips in transit depict a larger host star. However, habitability decreases with increase in these features.

Figure 4: Stellar Mass-Temperature relation with Habitability

The correlation matrix depicts the relationship between each numeric features which can be seen in Figure 5. There are positive and negative correlations. A positive correlation indicates if one variable increases, the other correlated variable tends to increase

whereas a negative correlation occurs when one variable tends to decrease with increase of the other variable. From the correlation figure, it can be inferred that the features P_ESI and P_HABITABLE have a positive correlation (0.371), suggesting as the Earth Similarity Index (P_ESI) of a exoplanet increases, the likelihood of it being habitable also increases. P_ESI and P_RADIUS have a strong negative correlation (-0.585) which suggests larger planets tend to have a lower Earth similarity index which is true. Understanding these relationships can help improve model interpretability and refine the dataset's analysis.

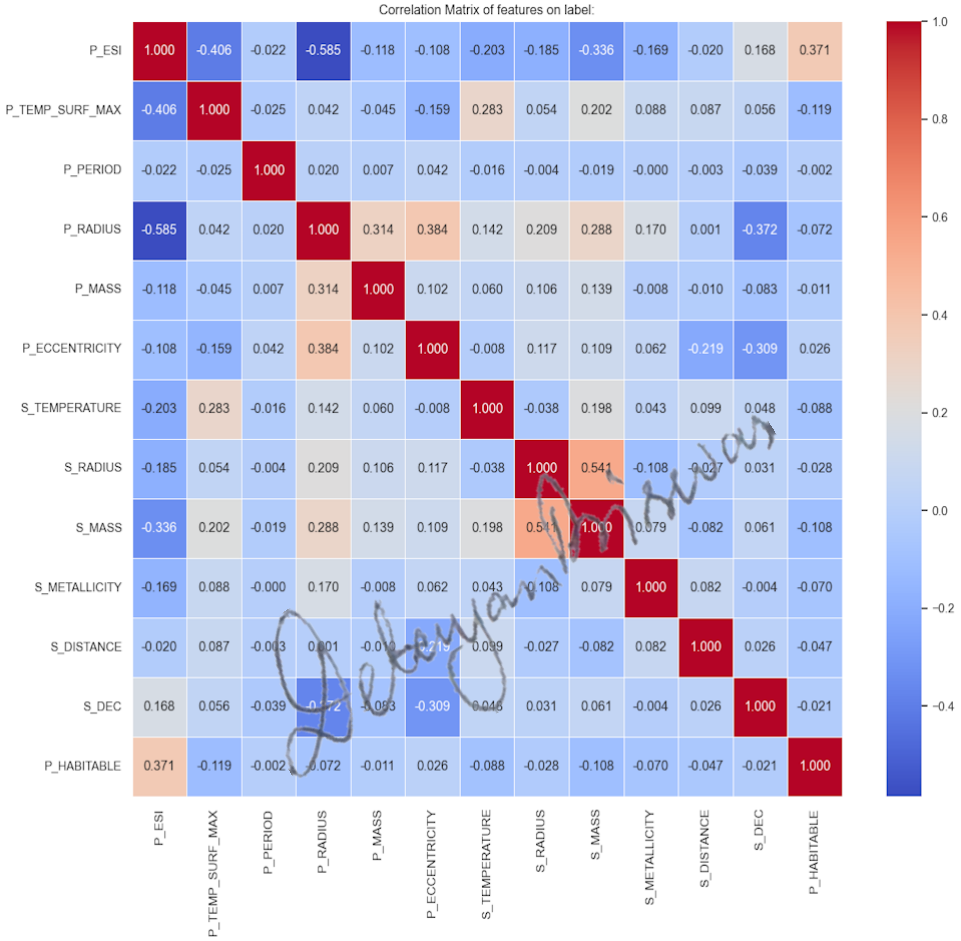

Figure 5: Correlation Heatmap of numeric features

3.5 CLASS IMBALANCE HANDLING

The merging of the two datasets brings in significant class imbalance with majority present under the majority class. The majority class belongs to non-habitable cases, is the major issue to address as it will impact the prediction efficiency. The count of classes 0, 1, and 2 are 5331, 22, and 39 respectively. To predict the habitability potential of exoplanets, there should be a balance between the classes present as it is necessary for the proper classification of the model. For this greater number of minority class samples needs to be generated and for this Conditional generative adversarial network (cGAN) is used. The cGAN block handles the dataset with the creation of two functions: Generator

and Discriminator. The generator function generates synthetic samples for the minority classes which are like the actual data samples. The discriminator class discriminates the synthetic sample by evaluating the authenticity of the synthetic data with the real data. The continuous training of the generator and discriminator creates a balanced dataset with an increase in the number of data generated. The cGAN algorithm generates realistic data for both minority class 1 and class 2. The synthetic data generating balances the dataset with each class having 5331 rows thereby increasing the dataset size to 15993 rows. The unnecessary categorical features are ignored because cGAN can only handle the generation of synthetic data based on the actual numeric data present and dealing with categorical features may lead to a misleading output. The P_TYPE_TEMP feature being categorical is transformed using Label encoder to generate numeric data based on cold, warm and hot categories. The features were scaled using Standard Scalar and the target label is encoded using One Hot encoder to ensure the data are in numpy array format as it is necessary for cGAN input.

3.6 MODEL TRAINING

The cGAN model output is evaluated based on the gradient boosting classifier XGBoost. XGBoost is used here for its known performance which has been discussed in Section 2 of this report. The target and features are first separated from the dataset to generate two unique tables. The features can be considered as the input to the machine learning algorithm and the target is the output generated which is the column P_HABITABLE. Both the features and labels are split into train and test sets with train and test size of 80% and 20% respectively . The features are scaled using Standard Scalar. This method standardizes the features by removing the mean ensuring all the features are equally considered for classification purposes irrespective of their actual scales. XGBoost being sensitive to the scale of input data makes this a necessary process. The target label is encoded using Label Encoder as it is categorical and has 3 categories as class 0,1 and 2. The classifier XGBoost is initialized for later training. Before that, hyperparameter tuning is applied to find the best optimal set of hyperparameters that can impact the model's performance. A grid search with cross-validation is implemented as hyperparameter tuning. A param grid is defined with the maximum depth of the tree for XGBoost, learning rate which controls overfitting, n_estimator which defines the number of trees to fit, subsamples, and features used for fitting each tree. After the parameters are defined, Grid Search CV is applied. A cross-validation of 5 is applied that splits the training data into 5 parts. The model is trained on the 4 parts and the remaining part is used for validation. This process is repeated 5 times with the validation set changing each time. This training gives the best hyperparameter as output to be used to evaluate the classifier.

4 Design Specification

This section consists of the algorithm architecture that is associated with this project. The algorithm proposed is a new combined approach that combines the power of cGAN with a powerful gradient boosting algorithm XGBoost. The details of the pseudo is discussed in this section.

Algorithm 1 Handle Class Imbalance with cGAN and Predict Exoplanet Habitability with XGBoost

Data Preparation:

Load **TESS** and **PHL** datasets
Preparing both datasets for merging
Merge both datasets on similar columns and specific columns
Pre-process the new dataset generated by handling numerical and catgeorical columns
Handling the missing values in habitable column using classifier
Dropping unnecessary categorical columns, and separate features and labels
Encode the *P_TYPE_TEMP* column into numeric values
Scaling features and encoding labels

Defining Generator Block:

generator *noise_dim, class_dim, output_dim*
Add Dense layer with 256,512 and 256 units using 'relu' activation
Add Dense layer with output_dim units using'sigmoid' activation
return model

Defining Discriminator Block:

discriminator *input_dim, class_dim*
Add Dense layer with 256, 128 and 64 units using 'relu' activation with Dropouts
Add Dense layer with 1 unit, 'sigmoid' activation
return model

Building cGAN model:

Defining input for noise_dim and class_dim
Generate data using generator with the inputs and set discriminator to False
Concat with class input and pass them through Discriminator for output
Define GAN model using output and inputs
Compile GAN using Adam Optimizer and loss function

Training cGAN:

steps **in** *num_epochs*
Select a random batch of real data and generate fake data using generator
Adding noise to prevent overfitting and use label smoothing
Train the discriminator on real and fake data
Train the generator through cGAN with real labels

Generating new Balanced data:

Calculating required synthetic samples for minority classes
Generating synthetic data for minority classes using generator
Combine real and synthetic data to create a balanced dataset

Training XGBoost Classifier:

Split balanced data into features and target label
Scale the features and encode the target label
Define parameter grid for XGBoost and use GridSearchCV to find best parameters
Train XGBoost model with balanced dataset

Evaluation of XGBoost Model:

Evaluate model performance using **Accuracy, Precision, Recall, Log loss, AUC-ROC curve**
Use XGBoost-specific KPIs like **SHAP, Feature importance**
Make habitability predictions using XGBoost model

The algorithm starts by fetching the two datasets that are required for the project. These datasets TESS and PHL, are then merged on similar columns and additional specific columns as these datasets consist of similar astronomical data. The merged dataset consists of both features and target label necessary for prediction. The preprocessing is done on the dataset to handle the missing values in the numerical and categorical columns. The missing values in the target label is handled using a RF classifier that predicts based on data patterns and relationship. The generator and discriminator blocks are defined as there will be implementation of GAN architecture. The generator takes noise and class label as input and generates fake data as output. The generator uses multiple dense layers with various activation function like relu and sigmoid, that determines how the sum of inputs is transformed into an output in neural network. The discriminator follows almost similar techniques on multiple layers with addition of dropouts. The difference between the two functions can be stated that the generator aims to generate realistic data, while the discriminator aims to distinguish real data from generated data. The cGAN block then combines the generator and discriminator as they are passed into the cGAN function. The cGAN model is compiled using loss function and Adam optimizer. The loss function determines closeness of prediction with actual data, whereas Adam optimizer adjusts the learning rate of model and loss function. The model training is done based on the necessary hyperparameters. The training process simultaneously trains the discriminator on real and fake data and trains the generator with real labels. The required minority classes are calculated and are generated using the trained cGAN model. The generated synthetic data from both classes 1 and 2 are combined with the real dataset to create a balanced dataset. The balanced features and labels are split into training and test set for the classifier. The features are scaled using Standard scalar and the target label is encoded using Label encoder. The hyperparameter grids are defined for the classifier and Grid search is used to find the best parameters. The XGBoost model is trained on the best found hyperparameters. The model's performance is based on quantitative metrics like accuracy, precision, recall, log loss and ROC-AUC curve. XGBoost specific qualitative metrics like Shapley additive explanations (SHAP) and feature importance are used to determine how each feature contributes to each prediction and identify the most important features. The model is then validated by passing a new exoplanet data that determines the class of the exoplanet accordingly.

5 Implementation

This section discusses the final implementation stages of the project that focuses on the model implementation, output generation, and tools and techniques used.

5.1 MODELS

The algorithm devised for this project is cGAN and its capabilities are extended using XGBoost classifier. The model is fine tuned to address the current problems of exoplanet habitability. The cGAN model is used to handle the imbalance in class present in the merged dataset. The generator block of the cGAN consists of a generator function that creates synthetic data like real data and discriminator function evaluates the authenticity of the generator data. The generator and the discriminator are trained simultaneously allowing the generator to improve its generative ability with every iteration and thereby

improving its ability to generate realistic synthetic data. The generator and discriminator are implemented using TensorFlow. The generator function takes random noise and class labels as input and generates data samples resembling the minority class. The discriminator on the other hand differentiates between the real and synthetic data samples. The discriminator uses Spectral normalization that is used to stabilize the discriminator training process ensuring smooth and reliable model training. Both the functions have dense and batch normalization layers with specific neurons. The cGAN model is build combining the generator and discriminator and is trained on the merged data using specific hyperparameters. Batch size, epochs, sample interval and noise_dim are specified as training parameters. The cGAN model generates synthetic samples for the two minority classes: Potential Habitable and Confirmed Habitable Class thereby creating balanced dataset. XGBoost is extended with the cGAN algorithm as a classifier. Its is a gradient boosting algorithm chosen for is performance in terms of accuracy and evaluation. XGBoost is implemented to train on the balanced dataset generated by cGAN and it thereby enhances the model's ability to accurately classify the classes.

5.2 RESEARCH RESOURCES

The researcher resources were significant for the successful implementation of this project.

5.2.1 Literature Paper

Research papers on machine learning techniques provided an idea of the proper logical approach to this project and helped in fine tuning the proposed approach. Articles and papers on class imbalance in dataset were important for understanding class imbalance handling and generation of synthetic data.

5.2.2 Documentation of libraries

The project needed the use of various libraries and packages for its successful implementation. The documentation of the libraries gave a proper understanding of the approach and helped in implementation of cGAN.

5.3 DATASET SOURCE

The datasets used for this project comes from two established sources: Transiting Exoplanet Survey Satellite (TESS) and Planetary Habitability Laboratory (PHL). The TESS and PHL both are available as open access and is free to use. TESS is available without licencing restrictions and PHL aligns with Creative Commons licence. The TESS dataset is maintained by the Mikulski Archive for Space Telescopes (MAST) funded by National Aeronautics and Space Administration (NASA) and includes data from Hubble, Kepler, TESS and other telescopes. The PHL dataset is maintained by the University of Puerto Rico at Arecibo.

- **TESS DATASET**- EXOPLANET ARCHIVE NASA
- **PHL DATASET**-PHL EXOPLANET CATALOG

5.4 TOOLS IMPLEMENTED

The implementation of the model involved the use of several libraries and software.

5.4.1 Jupyter Notebook

It is used as it provides and environment for testing the code. The notebook is also used to document the implementation and visualization of the result.

5.4.2 Key libraries

- **Pandas and NumPy:** Pandas is used for data manipulation and data processing for model training. NumPy is used for numerical computations and is needed for implementing algorithms.
- **Matplotlib, seaborn and missingno** Matplotlib is used for creating static and interactive visualizations. Seaborn is an extension of matplotlib used for creating complex plot and consist of numerous default style settings. Missingno is used for visualization of missing data in a dataset.
- **Scikit-Learn:** It is used for data preprocessing and also for evaluation of the model's performance.
- **TensorFlow:** It is used for creating, training and optimization of the cGAN model.
- **XGBoost:** It is used for implementing the XGBoost classifier which is important for final model evaluation.

5.5 HARDWARE IMPLEMENTED

The hardware used for this project includes the use of an AMD-based 6 core 12 logical core system with integrated Radeon Vega 7 graphics, 512 gigabytes of solid-state storage, and 16 gigabytes of RAM.

6 Evaluation

This section provides a complete analysis of the result obtained. The model's performance is evaluated based on a combination of quantitative and qualitative analysis to provide a complete assessment of the project.

6.1 Quantitative Analysis

6.1.1 F1 score and other metrics

The model evaluation gives an accuracy score of 0.95967 and this suggests the model correctly predicted the habitability class of exoplanets in 96% of the cases. Figure 6 shows the weighted average of precision and recall for the classes 0, 1, and 2 is at 0.96. The weighted F1 score of the model is 0.95968 as seen in Figure 7 suggests that the model performs well across all classes, balancing precision and recall efficiently.

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	1064
1.0	0.95	0.93	0.94	1108
2.0	0.93	0.95	0.94	1027
accuracy			0.96	3199
macro avg	0.96	0.96	0.96	3199
weighted avg	0.96	0.96	0.96	3199

Figure 6: Performance metrics

6.1.2 Confusion matrix

The confusion matrix in Figure 7 defines class 0 for the non habitable exoplanet class shows that 1064 instances are correctly classified with no misclassifications. The class 1 for potentially habitable class shows 1031 instances are correctly classified, but 77 instances are misclassified as class 2. The class 2 for confirmed habitable shows 975 instances are correctly classified, with 52 instances misclassified as Class 1. The evaluation depicts that the models has excellent precision for class 0 but experience small confusion for class 1 and 2.

6.1.3 Log Loss

Figure 7 shows the log loss obtained is 0.1010. Log loss measures the model's performance with a predicted output. A very low value of log loss indicates that the model suffered less during prediction and its predictive probability is close to the actual target label. The low value also indicates that the model can provide accurate probability estimates for each class.

6.1.4 AUC-ROC

The Area Under the Receiver Operating Characteristic Curve (AUC-ROC) score is 0.9957 as seen in Figure 7. This is a performance metric for classification problems at various thresholds. The score close to 1 depicts the model's ability to differentiate between different classes with a probability of 99.58%. This high score states strong model performance.

```
The final weighted F1 Score: 0.9596872922672619
Confusion Matrix:
[[1064    0    0]
 [   0 1031   77]
 [   0   52  975]]
The AUC-ROC Score: 0.9957563383427092
The Log Loss: 0.10103874634047534
```

Figure 7: Quantitative metrics

The ROC curve in Figure 8 shows high AUC values for the 3 classes. Class 0 having blue curve with AUC score of 1 indicates that the model can perfectly distinguish Class 0 from the other classes without errors. Class 1 and 2 both at 0.99 indicates robust performance with only a 1% of misclassification.

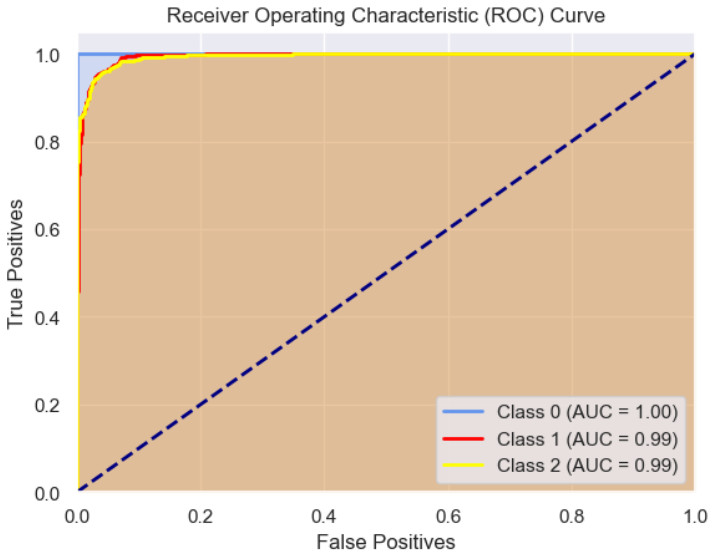

Figure 8: Receiver Operating Characteristic (ROC) Curve

6.2 Qualitative analysis

6.2.1 SHAP summary

The SHAP summary plot as seen in Figure 9 determines how different features aids the model prediction. The plot suggest that P_ESI, P_TEMP_SURF_MAX, and P_PERIOD significantly impact the model's prediction and also has significant interaction with the other features which can be observed in the plot as high interaction is plotted in red while low interaction in blue.

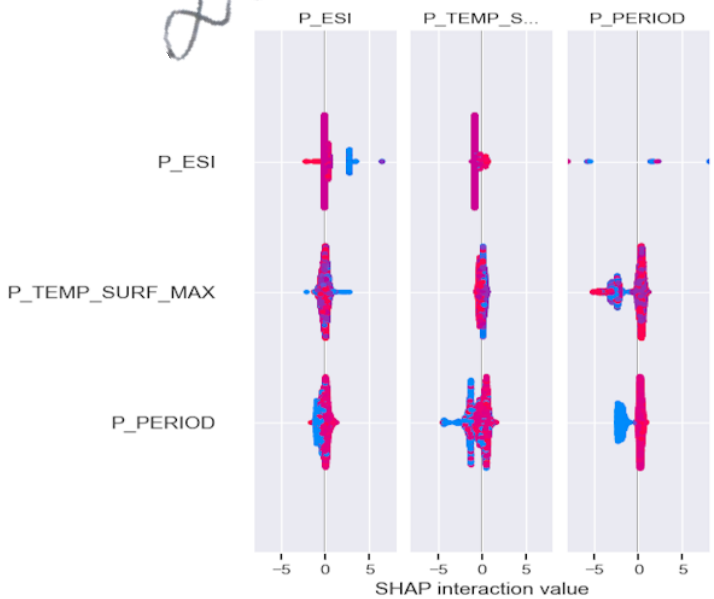

Figure 9: SHAP summary plot

6.2.2 Feature Importance

Feature importance ranks the features based on their contribution in model's prediction. The Figure 10 shows that P_PERIOD is the most significant feature with a score of 35.2072, indicating its crucial role in determining the habitability class of exoplanets. Higher values of the features positively influence the likelihood of a planet being habitable or potentially habitable.

	Feature	Importance
2	P_PERIOD	35.207203
6	S_TEMPERATURE	23.566950
12	P_TYPE_TEMP	10.072222
7	S_RADIUS	5.715146
8	S_MASS	3.867526
3	P_RADIUS	3.022836
1	P_TEMP_SURF_MAX	2.403355
0	P_ESI	1.779553
5	P_ECCENTRICITY	1.762819
11	S_DEC	1.329855
10	S_DISTANCE	1.185394
9	S_METALLICITY	1.148611
4	P_MASS	1.104955

Figure 10: Feature importance table

6.3 Validation check

New exoplanet data is passed as input into the model validation function to check its class predicting capability. The input data belongs to confirmed habitable planets. The output of the validation check function provided a result showing as confirmed habitable. This suggests the model is capable in categorizing the habitability of exoplanets based on input data.

6.4 Discussion

The model demonstrates strong performance with high accuracy and a weighted F1 score. The imbalance class was handled effectively by cGAN. P_PERIOD is the most influential feature that drives the predictions. The confusion matrix confirm the model's effectiveness in accurately classifying exoplanet habitability. The evaluation depicts the robustness of the model's performance in predicting exoplanet habitability.

7 Conclusion and Future Work

The main goal of this project was to address the research question, achieved through a process from data merging to model evaluation and validation. The major issue of class imbalance for this multiclass classification was handled effectively by the cGAN model generating very low D loss and G loss of 0.3855 and 1.5873 respectively. The model shows strong performance with an accuracy of approximately 96% and a weighted F1 score of 0.95968. A high ROC-AUC score of 0.9957 confirms the model's effectiveness in differentiating habitability classes. The confusion matrix shows that the model provides highly accurate predictions across all three classes. This suggests the project can aid researchers in identifying potential habitable exoplanets.

Future work could focus on various areas to improve model's efficiency and performance. Utilizing better quality dataset with more relevant features will likely improve the model's accuracy further. More efforts should be made to combine datasets from various sources to create a more enriched dataset that includes all necessary and additional features. Class imbalance handling with advanced and complex GAN algorithms and autoencoders could eventually improve the model's performance in predicting exoplanet habitability. Application of different combinations of classifiers and ensemble methods may yield more fruitful results. The knowledge obtained from this research could be used in predicting the habitability of other celestial bodies like asteroids and planetoids, and could broaden the scope and impact of this research. This project is not only limited to exoplanets detection but also can be implemented in other fields due to the flexibility of the techniques. The exoplanet research will donate a significant amount of knowledge into the understanding of the void and could pave the way in searching for life forms.

References

Bahel, V. and Gaikwad, M. (2022). A study of light intensity of stars for exoplanet detection using machine learning, 2022 IEEE Region 10 Symposium (TENSYMP), pp. 1–5.

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.

Chen, S.-A., Li, C.-L. and Lin, H.-T. (2024). A unified view of cGANs with and without classifiers, Curran Associates Inc., Red Hook, NY, USA.

Dai, Z., Ni, D., Pan, L. and Zhu, Y. (2021). Five methods of exoplanet detection, Journal of Physics: Conference Series **2012**(1): 012135.
URL: *https://dx.doi.org/10.1088/1742-6596/2012/1/012135*

Douzas, G. and Bacao, F. (2018). Effective data generation for imbalanced learning using conditional generative adversarial networks, Expert Systems with Applications **91**: 464–471.
URL: *https://www.sciencedirect.com/science/article/pii/S0957417417306346*

Ghaleb, F. A., Saeed, F., Al-Sarem, M., Qasem, S. N. and Al-Hadhrami, T. (2023). Ensemble synthesized minority oversampling-based generative adversarial networks and random forest algorithm for credit card fraud detection, IEEE Access **11**: 89694–89710.

Jakka, M. S. (2023). Assessing exoplanet habitability through data-driven approaches: A comprehensive literature review.
URL: *https://arxiv.org/abs/2305.11204*

Khoda, M. E., Kamruzzaman, J., Gondal, I., Imam, T. and Rahman, A. (2021). Malware detection in edge devices with fuzzy oversampling and dynamic class weighting, Applied Soft Computing **112**: 107783.
URL: *https://www.sciencedirect.com/science/article/pii/S1568494621007043*

Li, C., Su, Y. and Liu, W. (2018). Text-to-text generative adversarial networks, 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1–7.

NASA Exoplanet Archive (2024). Confirmed exoplanets table, Available online. Accessed: 2024-07-28.

Newman, P. D., Plavchan, P., Burt, J. A., Teske, J., Mamajek, E. E., Leifer, S., Gaudi, B. S., Blackwood, G. and Morgan, R. (2023). Simulations for planning next-generation exoplanet radial velocity surveys, The Astronomical Journal **165**(4): 151.
URL: *https://dx.doi.org/10.3847/1538-3881/acad07*

Planetary Habitability Laboratory (2024). Phl's exoplanets catalog, Available online. Accessed: 2024-07-29.

Prasad, M. S., Verma, S. and Shichkina, Y. A. (2023). Astronomical image processing: Exoplanet detection, 2023 XXVI International Conference on Soft Computing and Measurements (SCM), pp. 336–340.

Qiao-Yang, H., Shen-Wei, Z. and Liu, H.-G. (2023). The potential of detecting nearby terrestrial planets in the hz with different methods, Publications of the Astronomical Society of the Pacific **135**: 094401.

Rojas-Ayala, B. (2023). Twenty-five years of exoplanet discoveries: The exoplanet hosts.
URL: *https://arxiv.org/abs/2301.03442*

Sharma, V., Goel, S., Jain, A. K., Vajpayee, A., Bhandari, R. and Tiwari, R. G. (2023). Machine learning based classifier models for detection of celestial objects, 2023 3rd International Conference on Intelligent Technologies (CONIT), pp. 1–7.

Singh, J. and Singh, N. T. (2023). Analysis of exoplanet detection methods using machine learning and deep neural networks, 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS), pp. 1242–1247.

Vishwarupe, V., Bedekar, M., Pande, M., Bhatkar, V. P., Joshi, P., Zahoor, S. and Kuklani, P. (2022). Comparative analysis of machine learning algorithms for analyzing nasa kepler mission data, Procedia Computer Science **204**: 945–951. International Conference on Industry Sciences and Computer Science Innovation.
URL: *https://www.sciencedirect.com/science/article/pii/S1877050922008535*

Yang, L. (2020). Conditional generative adversarial networks (cgan) for abnormal vibration of aero engine analysis, 2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT, pp. 724–728.

Yi, H., Jiang, Q., Yan, X. and Wang, B. (2021). Imbalanced classification based on minority clustering synthetic minority oversampling technique with wind turbine fault detection application, IEEE Transactions on Industrial Informatics **17**(9): 5867–5875.