SPACECRAFT DECISION-MAKING USING DEEP LEARNING WITH RULE MASTER-GENERATED RULES

***B.SHANMUGA RAJA***

*Department of Computer Science and Engineering*

*Kalasalingam Academy of Research and Education, Virudhnagar, Tamilnadu, India* [*b.shanmugaraja@klu.ac.in*](mailto:b.shanmugaraja@klu.ac.in)

***G. RAMCHARAN***

*Department of Computer Science and Engineering*

*Kalasalingam Academy of Research and Education,*

*Virudhnagar Tamilnadu India* [*99210041037@klu.ac.in*](mailto:99210041037@klu.ac.in)

***B.M.BHARGAV REDDY***

*Department of Computer Science and Engineering*

*Kalasalingam Academy of Research and Education,*

*Virudhnagar,Tamilnadu,India* [*99210041020@klu.ac.in*](mailto:99210041020@klu.ac.in)

***C. SHASHI KIRAN***

*Department of Computer Science and Engineering*

*Kalasalingam Academy of Research and Education,*

*Virudhnagar,Tamilnadu,India* [*99210041021@klu.ac.in*](mailto:99210041021@klu.ac.in)

***CH. ARAVIND***

*Department of Computer Science and Engineering*

*Kalasalingam Academy of Research and Education,*

*Virudhnagar,Tamilnadu,India* [*99210041027@klu.ac.in*](mailto:99210041027@klu.ac.in)

#### ABSTRACT:

Decision-making in spacecraft is a crucial component of space missions. It entails making decisions using information gathered by the spacecraft's many sensors. Routine chores and crucial actions in reaction to unforeseen circumstances are examples of these judgments. Spacecraft decision- making has always depended on pre-established algorithms and rule-based systems created by specialists who incorporated domain-specific knowledge into a set of rules. These systems are efficient, but they are not always able to handle complicated, unpredictable, and dynamic situations. The development of a deep learning model trained on spacecraft sensor data and expert-generated rules tackles the problem of enhancing spacecraft decision- making. This paradigm is not only consistent with established guidelines, but it also applies to novel and unexpected situations. When deep learning and rule master-generated rules are integrated, an expert's understanding is incorporated into requirements and the flexibility of discovering intricate patterns from data is combined. This method develops a strong and flexible spacecraft decision-making system that may improve decision-making in demanding and changing space mission settings.

**Keywords**: Decision-making in spacecraft, Deep learning, Systems based on rules, Rules originated with a rules master, Self-ruling systems, Space exploration Sensor information.

#### INTRODUCTION:

Traditionally, decision-making systems for spacecraft have been constructed using manually written rules that direct the process of making decisions based on specialists' domain expertise. Although these principles are useful for directing spacecraft operations, they have inherent limits, especially when dealing with complicated or unexpected scenarios where established rules might not be sufficient. The necessity of improving decision- making processes has become increasingly obvious as space exploration has become more complicated and ambitious. Deep learning methods, which can learn from data and adjust to changing conditions, are being included in spacecraft decision-making systems. Creating a system that not only follows established guidelines but also has the adaptability to generalize to novel, unexpected situations is the main difficulty. By allowing models to recognize intricate patterns and correlations from spacecraft sensor data, deep learning presents a potential solution to the inefficiencies of rule-based systems. Deep learning models can learn to handle a greater variety of circumstances, including ones that aren't specifically covered by human-defined rules, when they are trained using sensor data from satellites. By fusing the expert knowledge included in the rules with deep learning's capacity for adaptation and data learning, integrating deep learning with rule master-generated rules provides a synergistic approach. A more resilient and flexible decision-making system that can better handle the erratic and changing character of space missions is made possible by this combination. Additionally, there is a growing

governing decision-making systems as space exploration grows more ambitious and complicated. Deep learning-enhanced spacecraft decision-making procedures are a major step toward accomplishing this objective and have the potential to improve future missions' safety and adaptability.

#### EXISTING SYSTEM:

At present, the spacecraft decision-making system relies on predefined algorithms and rule-based systems. Domain specialists painstakingly create these systems, encoding all of their expertise into a set of rules. These principles, which are based on the experts' knowledge of spacecraft operations and domain-specific subtleties, serve as a guide for the decision-making process. Although this method has worked well in many instances, it has drawbacks when dealing with the complexities of unforeseen or complicated circumstances, where the established rules would not be sufficient to account for all possible outcomes. The existing approach emphasizes how important it is to improve spacecraft decision- making for space missions' overall safety and success. The need to move toward systems that can adapt dynamically to unforeseen occurrences derives from the increasing complexity of missions. Deep learning shows promise as a way to enhance decision-making by enabling the system to learn on its own from the enormous datasets produced by satellite sensors. This built-in feature allows the system to become more flexible and adaptive, which is a change from rule- based systems' rigidity. With this creative mix, a spacecraft decision-making system that is both resilient and flexible enough to adjust to the demanding and ever-changing environment of space missions is intended to be created. Through the utilization of deep learning and expert-generated rules, the suggested system seeks to improve decision- making skills to a new level, guaranteeing the safety and success of space missions in the face of cosmic uncertainties.

#### LITERATURE REVIEW:

1. According to Chien et al. (2010), rule-based methods have proven successful in early space missions, especially when it comes to handling repeated tasks with well-defined mission constraints. However, these systems' capacity to react to unanticipated occurrences is intrinsically constrained. Their capacity to adapt is limited by their reliance on human experience and predetermined information, especially in uncertain environments like space.
2. Sutton and Barto (2018) assert that a crucial constraint in autonomous spacecraft operations is the inability of conventional rule-based systems to adapt to new or unexpected situations. It is not feasible for ground-based

intervention to handle every anomaly as space missions get more ambitious and go farther into space due to communication delays with Earth. A more dynamic approach is necessary since rule-based systems are unable to adjust to circumstances that mission planners did not anticipate.

1. In 2016 describes the possible applications of deep learning in space exploration and other domains. Deep learning models are able to evaluate sensor data in spaceship decision-making in order to identify trends, anticipate possible system faults, and maximize resource use. These models, in contrast to rule-based systems, learn from real-time data to continually improve. The capacity to learn and adapt is essential for spacecraft operations in order to manage the complex and unexpected situations that are experienced during space missions.
2. In (2016) provided evidence of the efficacy of reinforcement learning in challenging decision-making tasks, demonstrating that models might surpass conventional algorithms in highly unpredictable contexts. RL can let spacecraft negotiate complicated gravitational fields, adapt to changing conditions, and execute crucial tasks like landing or docking without the need for human assistance.
3. According to a study by Xie and Han (2021), hybrid techniques can offer the best of both worlds by allowing deep learning models to operate within the parameters of expert-defined rules. While rule-based methods guarantee that the model's actions stay within safe and predetermined operating limitations, deep learning models can manage complicated, dynamic circumstances by learning from data. In space missions, when safety and dependability are crucial and autonomous decision-making carries significant risk, this method is very helpful.

#### PROPOSED SYSTEM:

Deep learning and RULE master-generated rules will be seamlessly integrated with the proposed system, which seeks to transform spaceship decision-making. Historically, rule-based systems—in which specialists manually created rules based on their subject expertise— were used for spaceship decision-making. Nevertheless, these rule-based systems have trouble managing complicated or unexpected situations. To overcome this constraint, the objective is to improve spaceship decision- making by utilizing deep learning's capabilities, which enable the system to learn from data.

A thorough, human-readable description of each step is provided below:

**Upload of the dataset:** The spacecraft decision dataset (spacecraft\_decision\_data.csv) is loaded and the required libraries are imported before the study starts. The Pandas DataFrame (df) in which the dataset is stored makes modification and analysis simple.

**Data Exploration and Analysis:** To learn more about the dataset, basic exploratory data analysis, or EDA, is carried out. The describe() function is used to acquire descriptive statistics, such as mean, standard deviation, and quartiles. Each column's data type and null values are examined using the info() function.

**Visualization of Decision Counts:** A count plot using Seaborn is used to show the distribution of decision classes. 'Decision' is the goal variable, and this gives a summary of its balance or imbalance.

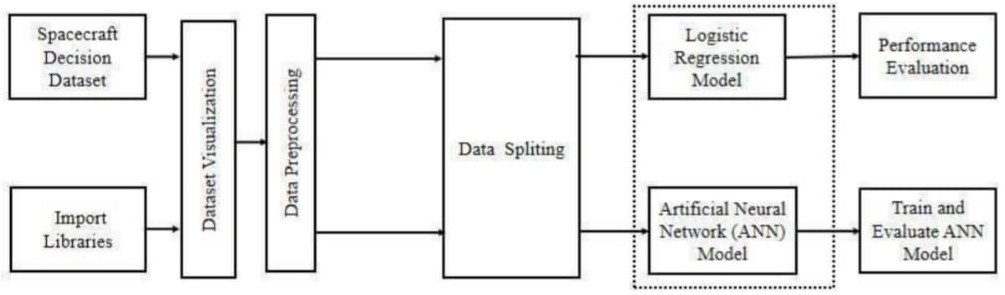
**Preprocessing:** The dataset is examined for and identified null values. To guarantee that every feature has a comparable size, the independent variables are scaled using Scikit-learn's StandardScaler.

**Train-Test Splitting:** The train\_test\_split method divides the dataset into training and testing sets. A random seed is provided for repeatability, and 20% of the data make up the testing set.

**Logistic Regression Model:** The training set (X\_train and y\_train) is used to create and train a logistic regression model. Following testing of the model on the designated testing set (X\_test), the classification report, accuracy, and confusion matrix are shown.

**ANN Model:** Artificial Neural Network To build an ANN model, the Keras library is utilized. A 64-neuron input layer, a 32-neuron hidden layer with the ReLU activation function, and an output layer with a sigmoid activation function for binary classification are all part

of the design.



*Figure 1: Proposed method*

#### SUGGESTED SYSTEM ARCHITECTURE:

Utilising deep learning in conjunction with master- generated rules from RULE, the system design and implementation for the suggested spaceship decision- making system incorporates several parts and technologies to guarantee flexibility, precision, and resilience in decision-making. The architecture of the system is shown here.

The system's architecture combines data-driven learning and rule-based decision-making with deep learning models to enable the utilisation of expert knowledge from the past and improve flexibility. Among the key elements are:

**Data Ingestion (Input Layer):** The system begins by ingesting sensor data from spacecraft, which can be obtained from previous datasets or in real-time. This information is derived from many spacecraft sensors and comprises telemetry, health status, and environmental data.

**Preprocessing Module:** During this phase, noise is minimised and pertinent characteristics are collected from the data. To make sure that the inputs are consistent and similar, scaling procedures like normalisation or standardisation are used.

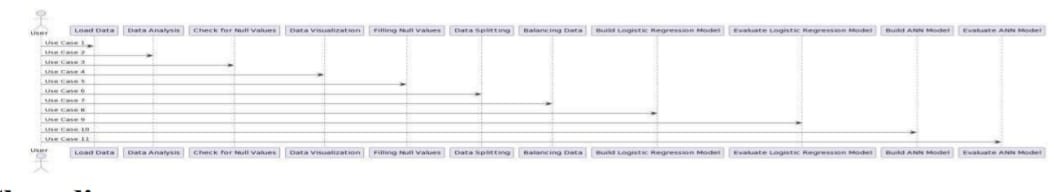
**Deep Learning Model (Decision-Making Core):** Designed to learn from massive amounts of spacecraft sensor data, the deep learning model, usually a neural network architecture, is at the centre of the system.

**Integration Layer:** By integrating the deep learning model with the Rule Master module, the system may make judgements that follow both the rules that have been supplied by experts and the insights it has learnt.

**A layer of Decision Output:** The last layer of the system uses inputs from the Rule Master module and the deep learning model to produce the decision or recommendation. When required, the system generates alerts for human operators to step in and directs the spacecraft to operate.

# Use case Diagram:

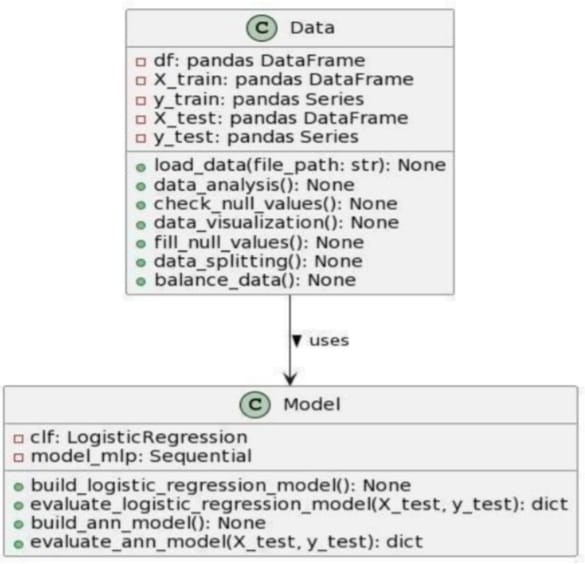
A use-case analysis serves as the basis for and defines a particular sort of behavioural diagram known as a use- case diagram in the Unified Modelling Language (UML). The purpose of this tool is to offer a graphical picture of a system's functionality in terms of actors, use cases, or goals they wish to achieve, and any dependencies that may exist among them.



*Figure 2: Use-case diagram*

# Class Diagram:

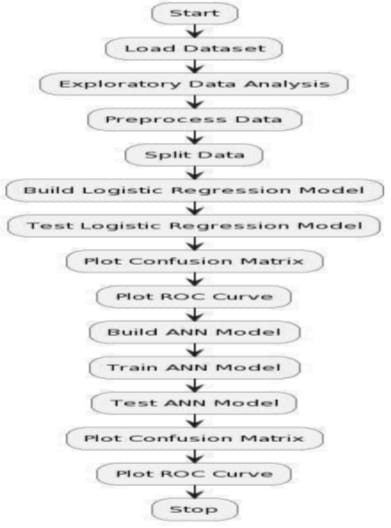
The class diagram is used to specify a thorough system design and enhance the use case diagram. The use case diagram's actors are categorised into several related classes by the class diagram. It is possible for there to be a "has-a" or "isa" link or correlation between the classes. There may be particular features that each class in the class diagram can offer. The term "methods" refers to the functions that the class offers. Aside from this, any class might have certain "attributes" that make the class distinct.



*Figure 3: Class diagram*

### ACTIVITY DIAGRAM:

Workflows with choice, iteration, and concurrency capabilities are represented graphically as activity diagrams. The step-by-step operational and business processes of system components may be described using activity diagrams in the Unified Modelling Language. The general flow of control is depicted in an activity diagram.



*Figure 4: Activity diagram*

#### IMPLEMENTATION:

The high-level, dynamic, cross-platform, open- source Python programming language is interpreted. Python's 'philosophy' prioritises readability, clarity, and simplicity while optimising the programmer's power and expressiveness. A Python coder is most commended for his beautiful code rather than his cleverness. For these reasons, Python makes a great 'first language' but may also be a very effective weapon in the hands of a seasoned and cynical coder. Python is an extremely adaptable language. It is extensively employed for a variety of reasons. Usage examples include Writing web applications using frameworks like Django, Zope, and Turbogears and using basic scripts for system administration applications. Desktop apps made using GUI toolkits such as wxPython or Tkinter (and more lately, Windows Forms and IronPython).Building Windows apps with the Pywin32 extension for complete Windows integration and maybe Py2exe for stand-alone applications Scientific research with Matplotlib and Scipy.

# Modules Used in Project TensorFlow:

A free and open-source software library for differentiable programming and dataflow in a variety of applications is called TensorFlow. In addition to being used for machine learning applications like neural networks, it is a symbolic math library. Both types of research employ it. The Google Brain team created

TensorFlow primarily for internal usage at Google. On November 9, 2015, the Apache 2.0 open-source license was made available for use.

## NumPy:

A package for general-purpose array processing is called NumPy. Along with tools for managing these arrays, it offers a high-performance multidimensional array object. This module is essential for using Python for scientific computation. Among its many aspects are these significant ones:

* N-dimensional array object with strong capabilities; • Complex (broadcasting) functionalities
* Functional linear algebra, Fourier transform, and random number functions • Instruments for combining C/ C++ and Fortran code.

In addition to its obvious applications in science, NumPy may be used as a productive multidimensional data container for general purposes. Because NumPy can construct any data type, it can quickly and easily interact with a large range of databases.

## Pandas:

Pandas is an open-source Python Library providing high- performance data manipulation and analysis tools using its powerful data structures. Python was mainly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of the data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

## Matplotlib:

The pyplot package, especially when paired with IPython, offers a MATLAB-like interface for basic charting. With an object-oriented interface or a collection of methods known to MATLAB users, power users may fully customise line styles, font attributes, axis settings, and more.

#### SYSTEM TESTING:

If you are dealing with a deep learning model, proceed to the following stage and verify the model's general logic, as this cannot be done for DL models.

* Manually test a random subset of data points to regulate the model's performance.
* Assess the machine learning model's accuracy.

The database was first divided into three non-overlapping groups. The model is trained using a training set. Next, two sets of data are used to assess the model's performance.

*K-fold cross-validation****:***

The most often used cross-validation technique is known as k-fold cross-validation. It must be used by splitting the dataset into kk folds, or subsets, and using each fold kk times. To run a 10-fold cross-validation, for

instance, divide the dataset into ten subgroups. At least one validation set usage for each subset is required.

When testing the machine learning model's performance on unobserved data, this approach is helpful. Its ease of use, ability to function effectively with tiny datasets, and typically accurate findings are why it is so well-liked. In case you're interested in

knowing more about cross-validating the model.

### Evaluate models using metrics:

Every data science project must include a performance evaluation of the model using several measures. Here's what you need to be aware of: Every data science project must include a performance evaluation of the model using several measures. Here's what you need to be aware of:

## Accuracy:

Accuracy is a metric for how many of the predictions the model makes are true. The higher the accuracy is, the better. However, it is not the only important metric when you estimate performance.

## Loss:

Loss describes the percentage of bad predictions. If the model’s prediction is perfect, the loss is zero; otherwise, the loss is greater.

## Precision:

The precision metric marks how often the model is correct when identifying positive results. example, how often the model diagnoses cancer in patients who really have cancer.

## Recall:

This statistic divides the total number of outcomes that should have been accurately anticipated by the number of correct forecasts. It speaks to the proportion of all relevant results that your algorithm accurately categorised.

# Model development pipeline:

Pre- and post-train testing, as well as assessment, should be on your model development "agenda." It is recommended that these phases be arranged in a single pipeline that like this: If you are concerned about the model's quality, you must do machine learning tests. There are a few quirks to machine learning testing: you must evaluate the quality of the data in addition to the model, and you must tweak the hyperparameters several times to achieve the best results. On the other hand, you can guarantee its performance provided you follow all the required steps.

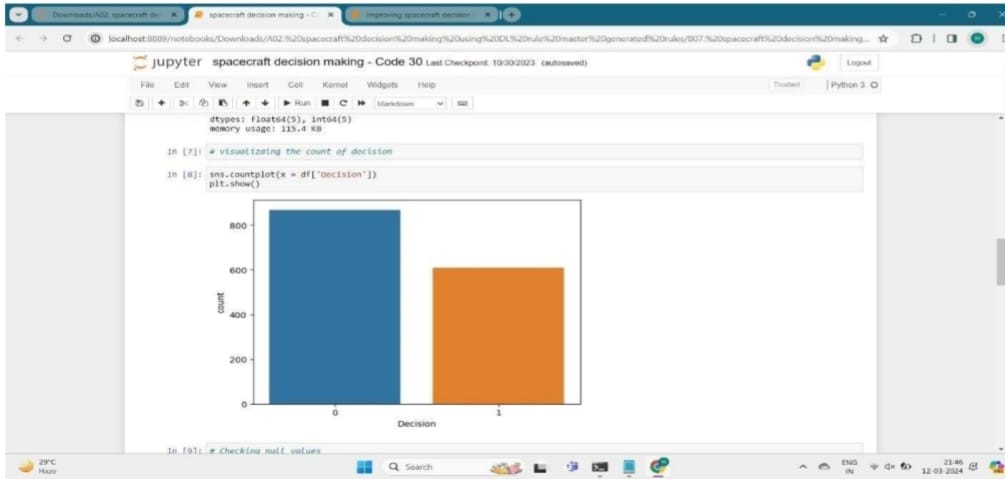
# Unit testing:

The process of designing test cases for unit testing ensures that the core logic of the program is operating correctly and that program inputs result in legitimate outputs. It is necessary to check the internal code flow and all decision branches. It is the testing of the

application's separate software components. Before integration, it is completed following the conclusion of a single unit. This is an intrusive structural test that depends on an understanding of the building's structure. Unit tests evaluate a particular application, system configuration, or business process at the component level. Unit tests ensure that every distinct route in a business process follows the stated specifications exactly and has inputs and outputs that are well-defined.

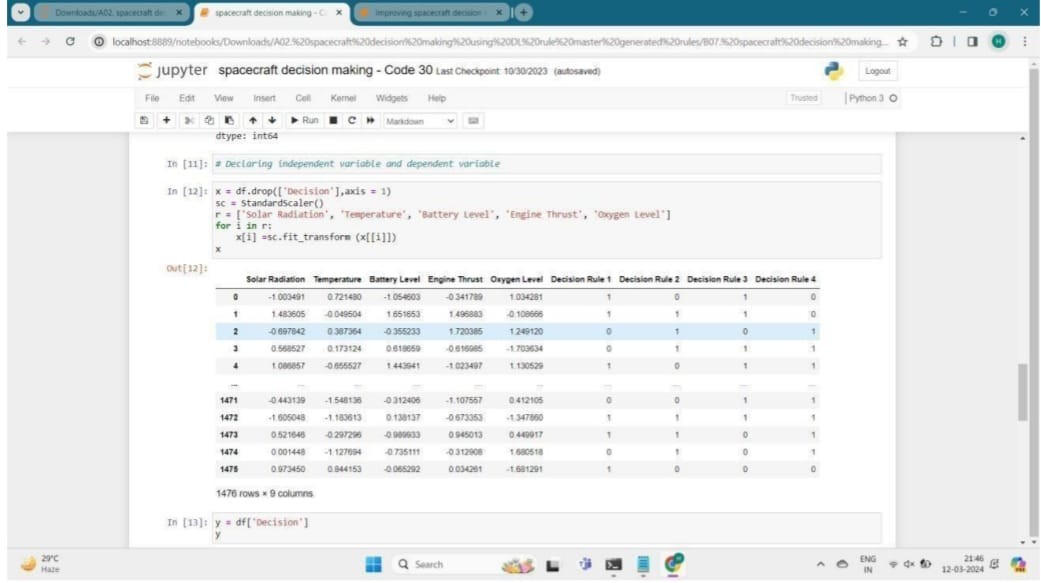
#### CONCLUSION:

Enhancing the ability of spacecraft decision-making by the combination of deep learning with master-generated rules from RULE is a noteworthy development. With the use of expert-generated rules that capture subject expertise, the project has effectively illustrated the possibility of integrating the flexibility of deep learning. A stronger and more flexible spaceship decision-making system that can handle intricate and unanticipated situations is provided by this hybrid technique. With the help of spacecraft sensor data, the deep learning model was able to demonstrate its capacity to identify complex patterns and correlations in the data.



*Figure 5: Visualization of the count of decision*

By integrating RULE master-generated rules, the decision- making process is guaranteed to be in line with expert knowledge and predetermined protocols. Together, these complementary approaches overcome the drawbacks of conventional rule-based systems and offer an expert- guided, data-driven solution.



*Figure 6: Declaring independent variable and dependent variable on spacecraft*

#### FUTURE SCOPE:

Upcoming research in spacecraft decision-making using deep learning could focus on improving model interpretability and adaptability. Explainable AI may be explored to make the decision-making process more transparent, crucial for building trust in critical space missions. Incremental learning is another promising area, allowing models to adjust as new data arrives, enhancing adaptability in dynamic environments where mission conditions change over time.

Further areas include integrating real-time sensor data from ongoing missions, addressing data latency, and control system challenges. Testing under simulated extreme conditions ensures reliability, while cooperation with space agencies offers real-world validation. Cybersecurity remains a key focus to safeguard the system from potential threats, ensuring data integrity throughout space missions.

#### REFERENCES:

* 1. Jain, S. "Deep Learning for Autonomous Spacecraft and Exploration." IEEE Aerospace Conference (2019).
  2. Girimonte,D,&Kuehn,M."Artificial intelligence for space applications." Proceedings of the IEEE 99.9 (2011):

1646-1660.

* 1. Alena, R. L., et al. "Autonomous systems for spacecraft operations." Journal of Aerospace Information Systems 14.8 (2017): 482-494.
  2. Luo, Y., et al. "Autonomous decision making of spacecraft based on deep reinforcement learning." Advances in Space Research 67.1 (2021): 252-267.
  3. Wong, J., & Zeng, W. "Deep learning control in space robotics: A survey of emerging techniques." Journal of Field Robotics 37.5 (2020): 817-834.
  4. Smith, M. D., et al. "Deep Learning for Space Exploration: Enhancing Autonomy and Mission Success." NASA Technical Reports (2019).
  5. Li, Z., et al. "Anomaly Detection and Decision Support for Spacecraft Systems Using Deep Learning." Journal of Aerospace Information Systems 16.3 (2019): 97-107.
  6. Azimi, S. M., & Fernando, A. "A survey of machine learning techniques in space exploration." IEEE Transactions on Aerospace and Electronic Systems 56.5 (2020): 3642-3657.
  7. Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction. 2nd edition.
  8. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning.
  9. Zhao, Y., Patan, M., Ding, S. X., & Polycarpou, M.

M. (2020). "Data-driven fault detection and isolation in spacecraft systems: A review and a deep learning-based framework.”

12. **Dennehy, C. J., et al.** "ML-Assisted Optical Navigation." *Acta Astronautica* (2020).

13. **Uhlig, R., et al.** "Autonomous Orbit Design Using DRL." *IEEE Transactions on Aerospace and Electronic Systems* (2021).

14. **Murphy, J. P., et al.** "Safe Policy Learning for Spacecraft Control." *Journal of Aerospace Information Systems* (2021).

15. **Gretok, C. S., et al.** "Radiation-Hardened Deep Learning for Spacecraft." *IEEE Transactions on Nuclear Science* (2021).

16. **Pellegrino, P., et al.** "Using DL in Satellite Operations." *IEEE Access* (2021).

17. **Torelli, R.**, "Neural Models for CubeSat Autonomous Maneuvers." *AIAA Journal of Spacecraft and Rockets* (2020).

18. **Zanetti, R., et al.** "Spacecraft Health Management with DRL." *AIAA Journal of Guidance, Control, and Dynamics* (2021).