# Metaverse in InterPlanet Internet: Designing a Space Robot That Can Make Self-Replication

Poondru Prithvinath Reddy



June 17, 2024

# Metaverse in InterPlanet Internet: Designing a Space Robot that can Make Self-replication

**Poondru Prithvinath Reddy**

**ABSTARCT**

The interplanet internet is a conceived computer network in space, consisting of a set of network nodes that can communicate with each other. These nodes are the planet's orbiters (satellites) and landers (e.g. robots, autonomous machines, etc.) and the earth ground stations, and the data can be routed through Earth's internal internet. As resource depletion on Earth becomes real, the idea of extracting valuable elements from asteroids or using space-based resources to build space habitats becomes more attractive, one of the key technologies for harvesting resources is robotic space mining ( minerals, metals, etc.,) or robotic building of space settlement. The metaverse is essentially a simulated digital environment mimicking the real world. The metaverse would be something very similar to real world planetary activities where users ( space colonies or internet users on Earth) interact with overlaying objects represented by robots, drones, etc. for real-world planetary activities like space mining, building space settlements, etc. in a completely virtual manner. Here we show how machines that use AI, which in turn make machines/ robots capable of using raw materials to reproduce itself by giving intelligence and collective knowledge to space robots in operation and also their interactions with the environment. Firstly, the self-replicating robotics is "conceived" by an artificial intelligence (AI) program working on group of robots and the AI ran an evolutionary algorithm capable of testing large number of robotic body shapes in a simulation. The goal is to discover which configuration of robots is capable of self-replication, in another words suitable for specific space application. The AI rendered a winning design by selecting a cluster of robots shaped like custom built designer model for space mining operation from the lot. Secondly, using the AI's blueprint and design features to hand-sculpt the robots, random robot parts added to the assembler site to give the parent robots raw material to make babies like their own bodies and the robobabies grow into parent robots. By adding robotic body parts, self-replication continues formation after formation and sculpting a robot shape out of robot-body parts is the "programming" that instructs body-parts clusters to develop a certain way. We implemented AI models such as Evolutionary Strategy and Neural Networks(CNN) in a replicating structure to classify robots and for robot cluster formation with sharing knowledge and intelligence. In this way, the desired response was measured, and new operational conditions were synergistically combined for desired outcomes. The results of the study show that the real individual behaviour on a distant planet of undertaking of space related activities with self-replication using evolutionary learning and programmed engineered design models could be of reality even in interplanet environment provided the interplanet internet is available as pathway communication.

## INTRODUCTION

Inter-planetary exploration, be it Lunar habitation, asteroid mining, Mars colonization or planetary science/mapping missions of the solar system, will increase demands for inter-planetary communications. The movement of people and material throughout the solar system will create the economic necessity for an information highway to move data throughout the solar system in support of inter-planetary exploration and exploitation. The communication capabilities of this interplanet information highway need to be designed to offer; 1) continuous data, 2) reliable communications, 3) high bandwidth and 4) accommodate data, voice and video.

The interplanetary Internet is a conceived computer network in space, consisting of a set of network nodes that can communicate with each other. These nodes are the planet's orbiters (satellites) and landers (e.g., robots), and the earth ground stations. For example, the orbiters collect the scientific data from the Landers on Mars through near-Mars communication links, transmit the data to Earth through direct links from the Mars orbiters to the Earth ground stations, and finally the data can be routed through Earth's internal internet. Interplanetary communication is greatly delayed by interplanetary distances, so a new set of protocols and technology that are tolerant to large delays and errors are required. The interplanetary Internet is a store and forward network of internets that is often disconnected, has a wireless backbone fraught with error-prone links and delays ranging from tens of minutes to even hours, even when there is a connection. In the core implementation of Interplanetary Internet, satellites orbiting a planet communicate to other planet's satellites. Simultaneously, these planets revolve around the Sun with long distances, and thus many challenges face the communications. The reasons and the resultant challenges are: The interplanetary communication is greatly delayed due to the interplanet distances and the motion of the planets. The interplanetary communication also suspends due to the solar conjunction, when the sun's radiation hinders the direct communication between the planets.

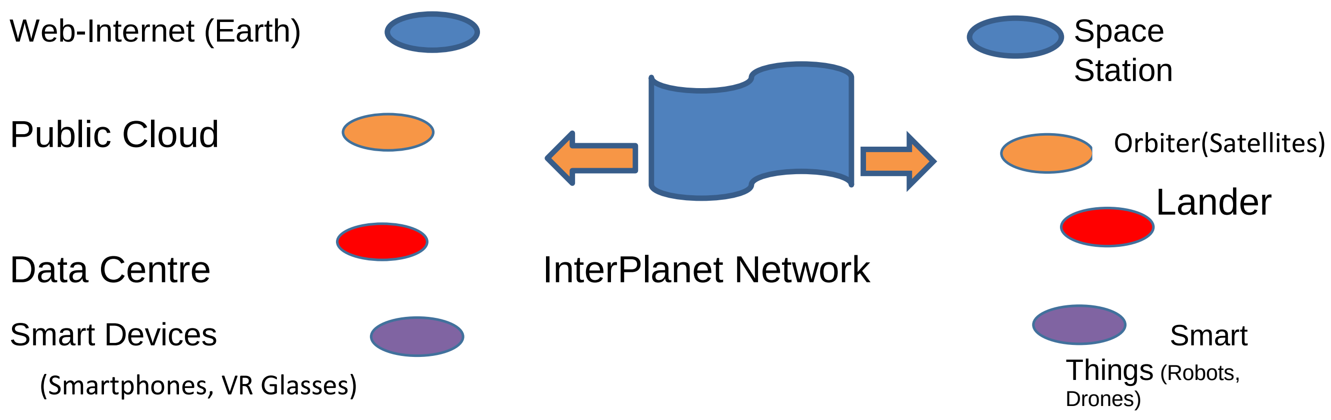
## NETWORK ARCHITECTURE

A **Computer Network Architecture** is a design in which all computers in a computer network are organized. An architecture defines how the computers should get connected to get the maximum advantages of a computer network such as better response time, security, scalability, etc.

Network architecture refers to the way network devices and services are structured to serve the connectivity needs of client devices.

- Network devices typically include switches and routers.
- Types of services include DHCP and DNS.
- Client devices comprise end-user devices, servers, and smart things.

The network architecture for the planet Mars or the Moon is as shown in below figure: -

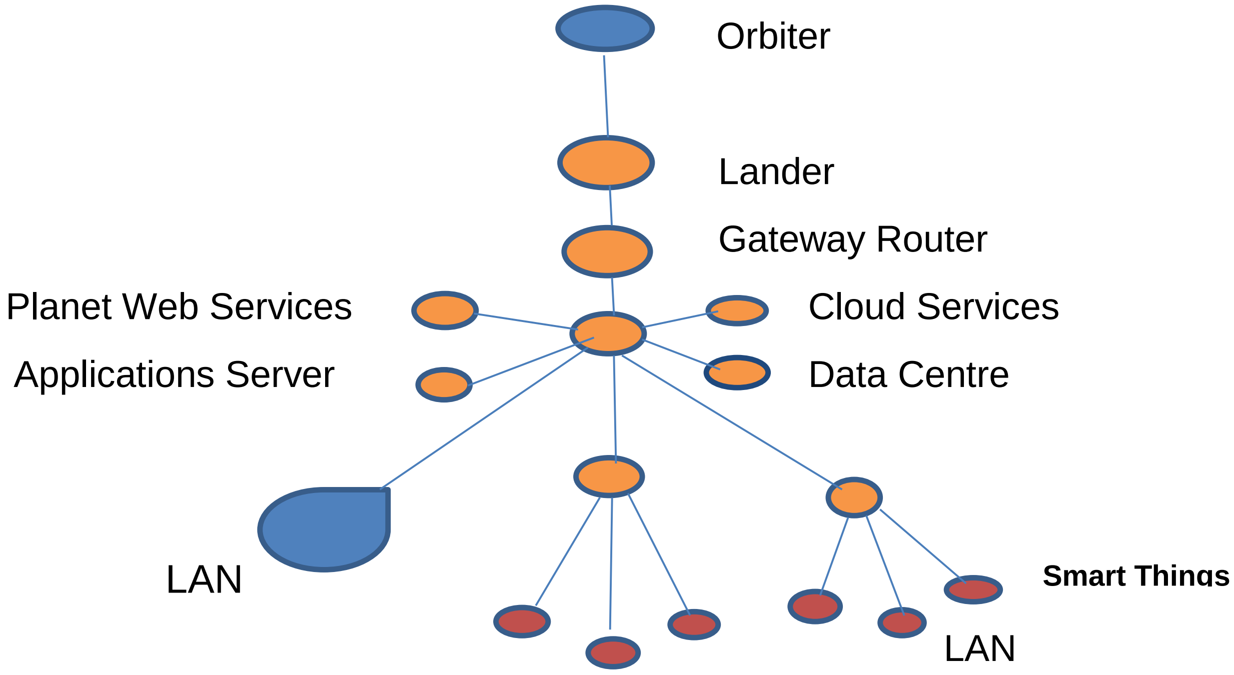



Computer networks are built to serve the needs of certain functionality and also their clients. Described below are three types of planetary networks:

- Access networks, for campuses and local areas, are built to bring machines and things onboard, such as connecting robots, drones, etc. within a location.

- Networks for data center connect servers that host data and applications and make them available to smart devices.
- Wide-area networks (WANs) connect robots and others to applications, sometimes over long distances, such as connecting robots to cloud applications related to space mining operations.

We give below the architecture of network on the planet Mars or the Earth's Moon is as shown in below figure: -




Local Area Network

An Internet is a "network of networks" in which routers move data among a multiplicity of networks with multiple admin. domains.

The main aim of networks is to connect remote endpoints with end-to-end principle and network should provide only those services that cannot be provided effectively by endpoints.

Since the networks are predominantly wireless, the fundamental impact of distance due to speed-of-light delays and impact on interactive applications – for both data and control is to be considered. Also power consumption of wireless links as a function of distance is to be examined.

The interplanetary internet is a conceived networks of nodes and these nodes are space station, planet's orbiters ( satellites ), planet's landers, robots ( drones, autonomous machines, etc. ), earth ground stations and earth's internal internet.

**METHODOLOGY**

### 1. *Augmented Reality*

The word ‘augmented’ means to add. Augmented reality uses different tools to make the real and existing environment better and provides an improved version of reality.

As Augmented Reality (AR) technologies improve, we are starting to see use cases and these include product visualization. There are AR apps that allow a customer to place virtual furniture in their house before buying and it is also a powerful tool for marketing as it allows users to try products before buying.

At its core, AR is driven by advanced computer vision algorithms that compares visual features between camera frames in order to map and track the environment. **But we can do more.** By layering machine learning systems on top of the core AR tech, the range of possible use cases can be expanded greatly.

Augmented Reality (AR ) can be defined as a system that incorporates three basic features: a combination of real and virtual worlds, real-time interaction, and accurate 3D registration of virtual and real objects

### 2. *Camera Representation*

A camera is a device that converts the 3D world into a 2D image. A camera plays a very important role in capturing three-dimensional images and storing them in two-dimensional images. And the following equation can represent the camera.

$$x=PX$$

Here x denotes 2-D image point, P denotes camera matrix and X denotes 3-D world point.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

The above is vector representation of x=PX [1].

The camera representation method is frequently used in image processing and is intended to identify the geometric characteristics of the image creation process. This is a vital step to perform in many computer vision applications, especially when metric information on the scene is needed.

### 3. *Metaverse Algorithm*

1. **Physical Reality Modeling -** required information
   - The goal of the agent/robot
   - What the robot sees, Materials & location
   - Real Simulation for Task Execution
2. **Task Execution ( Simulation )**
   - Generating actual materials ( how materials arrive at the site)
   - Robots arrive in the environment (speed and goal )
   - Task Execution ( Simulation Steps ), is updated as the work process progresses in line with the simulation
   - Task execution performance, as we have fully functional simulator and to make a realistic system, we would like to see how well it performs and mirrors real world execution( Artificial Intelligence )
   - Implementation of Graphical Version of the Task Execution

### Models for Metaverse & Algorithm

### Minimum amount of required information

- The current state of the robot/agent and its environment
- The goal of the agent/robot
- What the agent sees, materials & it's location

#### Agents – Attributes

We opt for the agents and they have the  attributes: the sight and the goal. While the goal is chosen randomly when an agent arrives on the location, the sight is always fixed to the some value. We define the autonomous robots as entities whose primary concern is to avoid failure; they should consequently not exhibit any preference for a certain speed as long as they are working safely. Furthermore, we add an attribute  to these learning agents; this is their probability of choosing a random action at each time step.

#### Agents as workmen

Given that we define learning agents the same way as the type of workers, we can seamlessly add them at the location. The only difference is how they will choose an action: by using their learning model, a neural network. We can therefore adapt the site's time step's algorithm to take the learning agent into account for the observation step. To decide what action it should take, the learning agent uses a neural network to approximate the Q-function. Thus, at every time step t, the agent c observes its state $s_{c,t}$; this state is then processed in some way so that it can be passed to a neural network.

#### Neural Network Models

Presently different neural network models are available that we will use to train our autonomous robots. These models define what information the learning agents use and how they are encoded as inputs to the neural networks. Before we start with our model, we need to define the building structure; how these neural networks are used by the learning agents. We use a feedforward neural network whose outputs correspond to the possible actions. Our models define different ways of using information about the agent's current state.

#### Required Information

We start by defining the minimum amount of information that an autonomous robot should have. Consequently, the model that we design will possess these pieces of information. They are:

- The goal of the agent/robot
- What the robot sees, Materials & location
- The current location that the agent is in
- The current speed of the agent
- Real Simulation for Task Execution

#### Task Execution ( Simulation )

- Generating actual materials ( how materials arrive at the site)
- Robots arrive in the environment ( speed and goal )
- Task Execution ( Simulation Steps ), is updated as the work process progresses in line with the simulation
- Task execution performance and to make a realistic system, we would like to see how well it performs and mirrors real world execution ( Artificial Intelligence with learning algorithm)

### Robotic Design and Placement

Robots are **collections of task executors** and have no brain system of their own. But they can be programmed to work autonomously and collaborate with other robots, or eventually to do other things tackling everything from space mining to deep space exploration.

Robots can be programmed as specific  executor of an assigned task for a number of situations and also using artificial intelligence to figure out the best shape for the Robots to perform in group on a more consistent basis to have better control over performance of assigned work.

Using a computational model that simulates the nature of work and everything of the Robot Capability, the process yields the robotic shape best suited to ensure the shape of the actual Robots into more efficient form suitable to a particular situation/task and accordingly enables robots to gather together in their environment forming them into groups with the same capability.

The revolution of modern computing has been largely enabled by remarkable advances in robotics however, majority of today's robots designed are not suitable for high-end space exploration, resulting in the need to speculate about how to optimize the next generation of robots for the machine learning (ML) models. Further, dramatically shortening the robot design/shape requirement would allow hardware to adapt to the rapidly advancing field of ML. The ML itself could provide the means to the robot design/shape requirement , creating a more integrated relationship between space exploration and ML.

Vast arrays of robots with different make are required for complex space applications thus, improving the selection of design patterns of these autonomous robots would be critical in improving the performance and efficiency of remote space applications and use of AI to achieve high-performance execution and robotic performance relevant to the work.

In order for the AI to design with an RL agent and the technique proved that AI can not only learn to design robotic patterns from scratch but that those structural patterns are accurate and faster than designed using any of the latest validation tools. Here an AI agent could design neural graphs and such a graph is converted into a class of robots with connection (relevant shapes) using a link generator.

**Collaborative Robotics**

Robots can be shaped to perform specific tasks. Robots have been designed and shaped in such a way that they can walk, swim, push pellets, carry payloads, carry shovelling and work together in a group to aggregate debris scattered along the surface  into neat piles or possibly, to build a space settlement. They can survive for long-time without recharge and heal themselves after any damage/confusion. The shape of a robot's body, and its distribution of legs and structure, are automatically designed in simulation to perform a specific task, using a process of trial and error.

Robots were able to move on their own. And using artificial intelligence, these  robots can be programmed as specific  executor of an assigned task for a number of situations and also using artificial intelligence to figure out the best shape for the Robots to perform in group on a more consistent basis to have better control over performance of assigned work.

Using a computational model that simulates the nature of work and everything of the Robot Capability, the process yields the robotic shape best suited to ensure the shape of the actual Robots into more efficient form suitable to a particular situation/task and accordingly enables robots to gather together in their environment forming them into groups with the same capability.

**Genetic Algorithm – Algorithmic Predictions**

The efficacy of space systems i.e. assessing parameters for improving space operational requirements could rely on a considerable number of variables and the amount of work required for exploring variables toward various elements in space systems can be reduced by using computational design approaches. Therefore, there is a need to  use a learning loop oriented by models generated from Computational Design approaches for optimizing space operational systems and achieving good accuracy in the combinatorial space calculation to be tested. And this requires testing of  (i) a combination of genetic designs of constructs by varying  around site requirements  and sequences, and (ii) other part components. By optimizing one variable at a time with a learning algorithm, the designs could managed to reach higher than that of normal increase in output in different iterations of execution.

Mathematical models either stochastic or deterministic can be digitally be evolved to generate optimal synthesis designs that satisfy a particular objective and  genetic algorithm is suitable for use to design shape regulatory platforms that exhibit ideal operational synthesis. A genetic algorithm is a metaheuristic inspired by the process of natural selection that belongs to the larger class of biological evolutionary algorithms. This algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring of the next generation. Initially a pool of unique designs were required to be generated from basic systems that satisfy design goal. These design courses were subsequently evolved using numerical simulations to obtain a desired output by repeated rounds of digital mutations and functional screening. These designs could serve as alternatives to consider, model or test during the operational synthesis cycle.

## ARCHITECTURE

### Robotic Shape Engineering

Robots are **collections of task executors** and robots can be shaped to perform specific tasks. Robots have been designed and shaped in such a way that they can walk, swim, push pellets, carry payloads, carry shovelling and work together in a group to aggregate debris scattered along the surface into neat piles or possibly, to build a space settlement. These robots can be programmed/shaped – to blend/ assemble other body parts, eventually to form into large robotic bodies. The parts required to be shaped in such a way that they gather other body parts in their environment forming them into new/ larger assemblies with the same capability/structure. And if there's enough of these part types in a container/site, they will start to assemble and will compact together.

The shape of a robot's part body and it's distribution/structure are automatically designed in simulation to perform a specific task, using a process of trial and error( an evolutionary algorithm ). Artificial Intelligence has been used to figure out the best shape for the robot structure.

Using a computational model that simulates the nature of application and everything of the robot capability, the process yields the robot shape best suited to ensure the shape of the actual robotic parts into more efficient form suitable to a particular situation/task and accordingly enables parts to compact together in their environment forming them into larger units with the same capability.

### Evolutionary Algorithm

Evolutionary Algorithms' (EA) constitute a **collection of methods** that originally have been developed to solve combinatorial optimization problems. Nowadays, Evolutionary Algorithms is a subset of Evolutionary Computation that itself is a subfield of Artificial Intelligence / Computational Intelligence.

An evolutionary algorithm (EA) is an algorithm that uses mechanisms inspired by nature and solves problems through processes that emulate the behaviors of living organisms.

The main classes of EA in contemporary usage are (in order of popularity) **genetic algorithms (GAs), evolution strategies (ESs), differential evolution (DE) and estimation of distribution algorithms (EDAs).**

Evolutionary algorithms are based on **concepts of biological evolution**. A 'population' of possible solutions to the problem is first created with each solution being scored using a 'fitness function' that indicates how good they are. The population evolves over time and (hopefully) identifies better solutions. Although genetic algorithms are the most frequently encountered type of evolutionary algorithm, there are other types, such as Evolution Strategy and we have chosen evolution strategy as preferred option as it suits our requirement of body shape engineering.

**Evolution Strategy**

In Evolution Strategy, we do not care much about the function and its relationship with the inputs or parameters. Some large population of numbers ( parameters of the model ) go into the algorithm and it squeeze out one value. We try to find the best set of such numbers which returns good values for our optimization and search problem.

The implementation process is as below:-

1. Randomly generate parameters ( population species )
2. Generate a population with different parameter vectors by sampling with Gaussian distribution
3. Evaluate each candidate by running the model and based on the output value evaluate the loss or the objective function by binding it to Evolution Strategy.

4. We then select top/best performing elite parameters and take the mean of these parameters for finding our best parameter so far
5. We then repeat the above process by again generating different parameters by adding Gaussian distribution to our best parameter obtained so far and repeat the whole process till best solution is found

**AI in Design**

Robot parts ( which are collections of body parts ) can be shaped to perform specific tasks. These parts can be programmed/shaped – to blend/ assemble other body parts, eventually to form into large robot bodies. The parts required to be shaped in such a way that they gather other body parts in their environment forming them into new/ larger units with the same capability/structure. And if there’s enough of these part types in a container/site, they will start to develop and will compact together.

The shape of a robot’s body and it’s distribution/structure are automatically designed in simulation to perform a specific task, using a process of trial and error.

AI contributes significantly to design thinking by offering **advanced data analysis, pattern recognition, and predictive modeling capabilities**. These technologies enable professionals to gain deeper insights, automate repetitive tasks, and generate innovative solutions at a faster pace.
When we're talking about artificial intelligence, however, pattern recognition is the technology that matches incoming data with information stored in a database. Thus, pattern recognition is a type of **machine learning** since it uses machine learning algorithms to recognize patterns

**AI Prediction Model**

AI Builder prediction models **analyze patterns in historical data that we provide**. Prediction models learn to associate those patterns with outcomes. Then, we use the power of AI to detect learned patterns in new data, and use them to predict future outcomes.
**Therefore, AI can increase model accuracy and broaden the use of predictive analytics applications** across industries to support intelligent and automated decision-making. AI and predictive analytics applications improve each other's capabilities and decision-making.

**Essentially, with predictive programming, we collect historical data, analyze it, and train a model that detects specific patterns so that when it encounters new data later on, it's able to predict future results. There are different predictive models that we can build using different algorithms. Popular choices include regressions, neural networks, decision trees, K-means clustering, Naïve Bayes, and others.**

**Predictive Modelling Applications**

There are many ways to apply predictive models in the real world. Most industries use predictive programming either to detect the cause of a problem or to improve future results.

Vast arrays of robots with different make are required for complex space applications thus, improving the selection of design patterns of these autonomous robots would be critical in improving the performance and efficiency of remote space applications and use of AI to achieve unprecedented acceleration for AI, high-performance execution, and robotic performance relevant to the work execution.

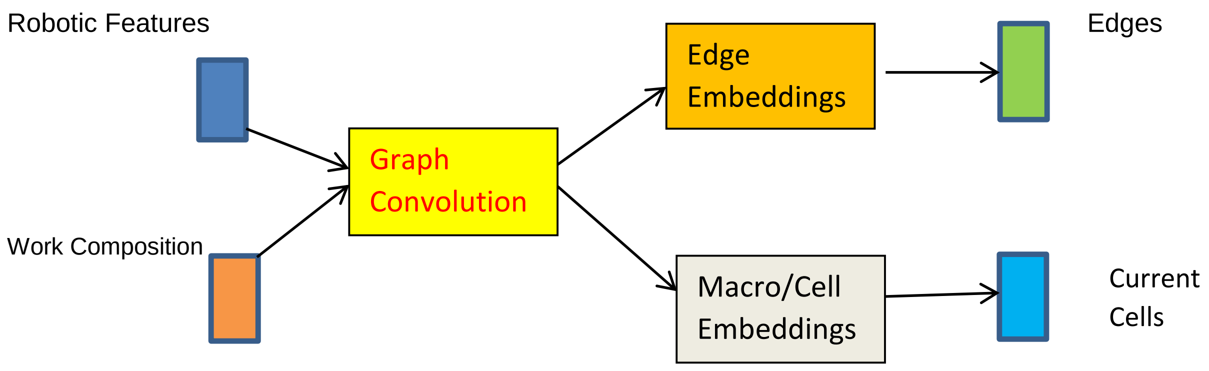
In order for the AI to design with a run at RL agent and the technique proved that AI can not only learn to design robotic patterns from scratch but that those structural patterns are accurate and faster than designed using any of the latest design/validation tools. Here an AI agent could design neural graphs and such a graph is converted into a class of robots with connection and 'relevant shapes' using a link generator. These generated circuits/graphs are then further optimized by a physical synthesis tool using synthesis optimizations such as robot sizing, etc.

Robotic types for task execution are built using logic and a lot of classifications/connections, should be easy, fast to reduce any delay that can be a drag on performance and consume as little power as possible, and the focus is on the size of the classification and the speed (for reducing delay).

The robotic type design is represented as a reinforcement learning (RL) task, where we train an agent to optimize the design and delay properties of robotic batch selection and for this relations are represented using grid representation with each element in the grid mapping to a graph node, and design an environment where the RL agent can add or remove a node from the connected graph.

We propose robot placement as a reinforcement learning (RL) problem, where we train an agent (i.e, an RL policy) to optimize the quality of robot placements. Unlike other methods, this approach has the ability to learn from past experience and improve over time. In particular, as we train over a greater number of robotic blocks, the method becomes better at rapidly generating optimized placements for previously unseen robotic blocks, and can rapidly generate optimized placements for space robots, and these methods can be applicable to any kind of robotic design patterns.

A fleet of robots are divided into dozens of blocks, each of which is an individual robotic module, such as a memory subsystem, compute unit, or control logic system and these blocks can be described by a graph of class components consisting of node types and graph adjacency information. The graph of type components/robots representing the composition and structural patterns, are passed through an edge based graph neural networks to encode input state. This generates the embeddings of the placed graph and the candidate nodes.




A graph neural network generates embeddings that are concatenated with the basic work meta data to form the input to the policy and value requirement of robotic design patterns for space exploration. The policy network generates a probability distribution overall possible grid cells onto which the current node/cells required could be placed for task execution.

## RESULTS

### Evolutionery Algorithm

Based on population of sample candidate solutions, and using  visualization library that shows how the accuracy changes across each population. It is observed that after few iterations, on the  simulated dataset, we are able to  find good accuracy . However, using higher values for the population of parameters might increase the accuracy. Therefore, evolutionary model as an optimization and shape engineering that mimics the concept of natural evolution / natural selection is encouraging.

**Neural Network Model**

**In obscene of graph databases using graph representation for machine learning systems** for managing robotic generation data, we build and store the graphs in a simple read format i.e. matrix representations ( stored as a node or record with edge list ) to perform link prediction.

We have represented this model as matrix with encoded values with possible values for each of the nodes along with the link attributes. We populated the matrix data with randomly generated data and simulated to represent the real world robotic elements/nodes as a link.
The system with different configurations for the hidden structures of the networks:

- 2 hidden layers: the first with 30 neurons and a *tanh* activation function; the second with 15 neurons and a *linear* activation function. No dropout.
- 2 hidden layers: the first with 30 neurons and a *tanh* activation function; the second with 15 neurons and a *linear* activation function. Dropout rate of 0.5.

The dropout rate of 0.5 has been chosen because it seems to be optimal for a wide range of networks.
The results for our CNN based model – RL policy model – The networks that do not use dropout seem to learn well. The percentage of desired generation for the networks (without dropout) is high.
Although we only have partial results, we can make the following observations: the networks that do not use dropout seem to learn well, while the network using dropout does not; it either learns very slowly or just converges to very low level of generation requirements.

## CONCLUSION

The interplanetary computer network in space is a set of computer nodes that can communicate with each other. We proposed a network architecture with planet's orbiters, landers (robots, etc.), as well as the earth ground stations and linked through Earth's internal internet, and consisted of complex information routing through relay satellites. As we know, the metaverse will be very different from the internet of today due to massive parallelism, three-dimensional (3D) virtual space and multiple real-world spaces like space mining, building space habitats, etc. We presented a self-replicating model of robotic deployment equipped with AI-model driven learning and programming that can effectively explore and execute complex applications in space environment and also designed an AI model of replication in two ways to address space related applications. As a part of the self-replicating robotics, (AI) program working on group of robots to discover which configuration of robots is capable of self-replication and the AI ran an evolutionary algorithm capable of testing large number of robotic body shapes in a simulation. Secondly, using the AI's design features to hand-sculpt the robots, random robot parts added to the assembler site to give the parent robots raw material to make babies and programming that instructs body-parts clusters to develop a certain way. Shaping a cluster of body parts in to specific configuration and then programming them to become a new self-replicating form of sharing of knowledge/ intelligence for working together. We implemented AI models such as Evolutionery Strategy and Neural Networks(CNN) in a replicating structure to classify robots and for building robot clusters with sharing knowledge and intelligence for successful execution. In this way, an AI - model assisted replication system with programming for sharing and collaboration that is part of Metaverse is feasible for automated execution of diverse space related outcomes and in that respect an implementation of Evolutionery strategy and Neural Networks(CNN) based on small model is presented. Although the platform model with AI learning and collaborative framework with replication structure given us a method of optimizing space applications however, this need to be tested using natural allocation for real space applications.

## REFERNCE


1. **Poondru Prithvinath Reddy**: **"Metaverse in InterPlanet Internet: Modeling, Validation, and Experimental Implementation", Google Scholar**