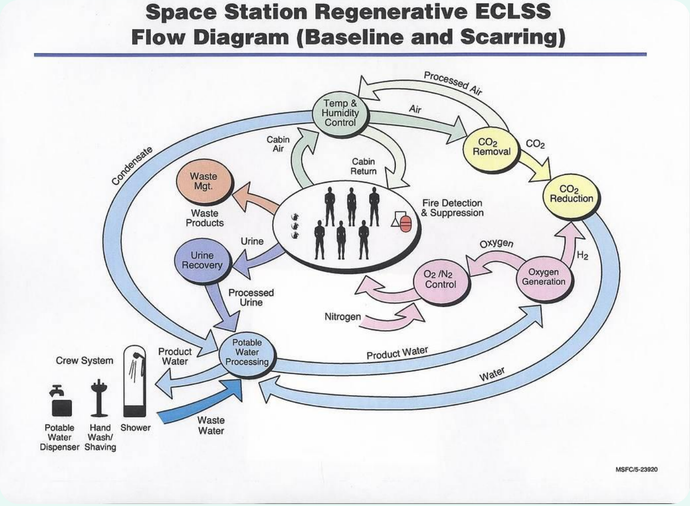
Implement an AI Preventive Maintenance System NASA ECLISS

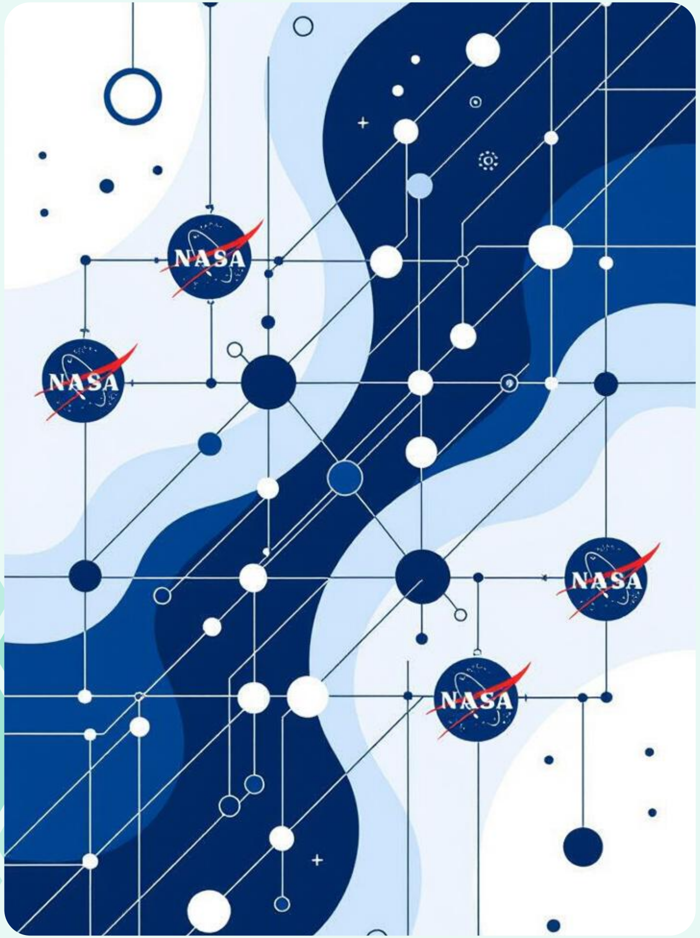


AI & ML for Predictive Maintenance of the NASA ECLSS System: Leveraging Real-Time Subsystem Logs, MCA Feedback, and Sensor Data

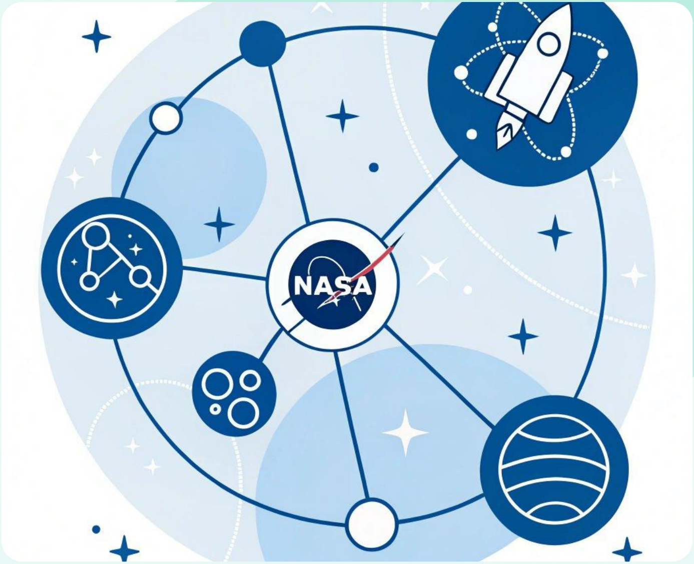
Purpose

To design and prototype an AI/ML-based predictive maintenance system for NASA's Environmental Control and Life Support System (ECLS- S).

- To utilize real-time logs from all ECLSS subsystems, feedback from the Major Constituent Analyzer (MCA), and other relevant sensors and alarm systems.



Problem Statement

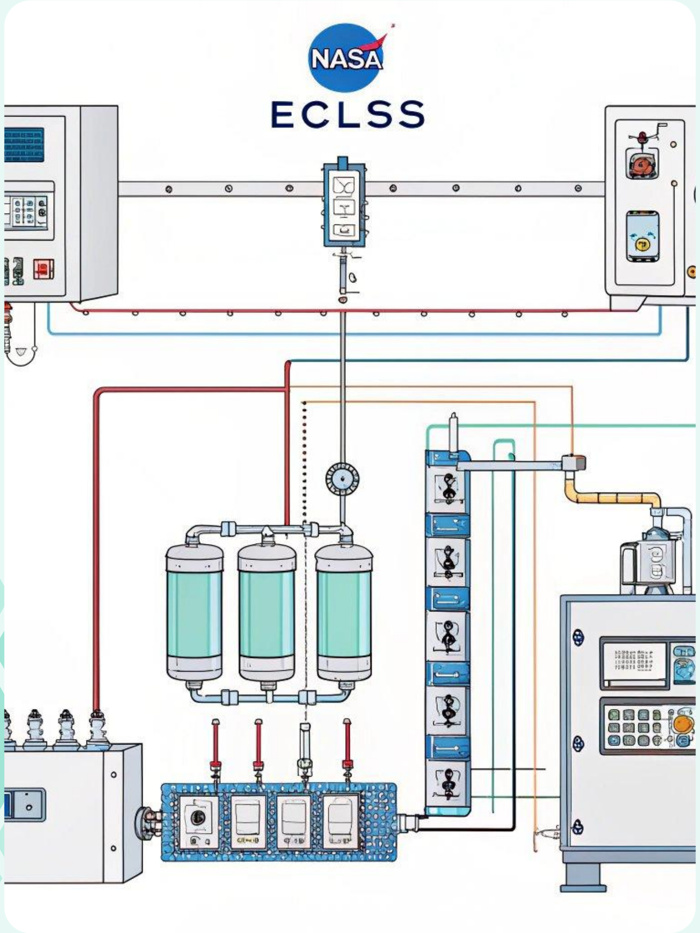


- The ECLSS is a complex, hierarchical system responsible for maintaining air, water, and waste management aboard spacecraft, such as the ISS.
- Unplanned failures in ECLSS subsystems can jeopardize crew safety and mission success.
- Current maintenance is largely scheduled or reactive, which can lead to inefficiencies, increased costs, and potential safety risks.
- There is a need for a predictive maintenance solution that can analyze real-time and historical data to forecast failures and optimize maintenance schedules.

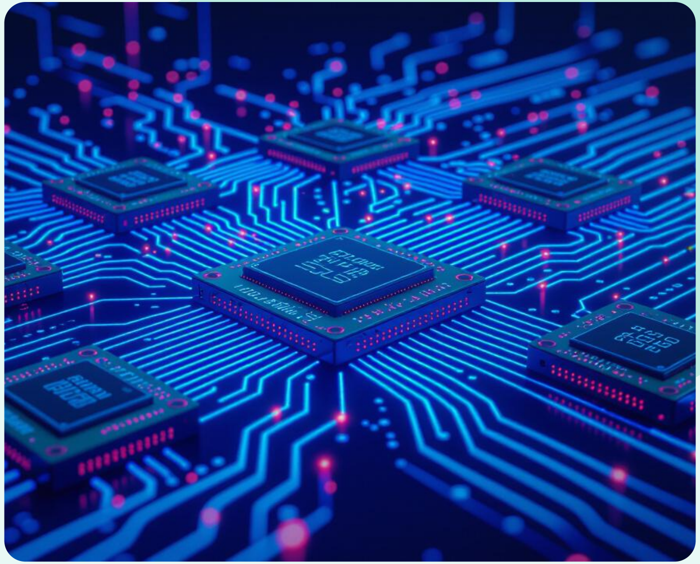
Prototype Hardware Requirements

Data Acquisition Hardware:

- Interfaces for collecting real-time logs from ECLSS subsystems (e.g., water recovery, air revitalization, oxygen generation).
- Integration with the MCA quadrupole mass spectrometer for atmospheric constituent data.
- Additional environmental sensors (temperature, humidity, CO_2, O_2, pressure, etc.).
- Alarm system interfaces for capturing event and anomaly notifications.



Prototype Hardware Requirements (Cont.)



Processing Hardware

- Edge computing device (e.g., Raspberry Pi, NVIDIA Jetson, or similar) for local data processing and ML inference.
- Network connectivity for data transfer and remote monitoring.

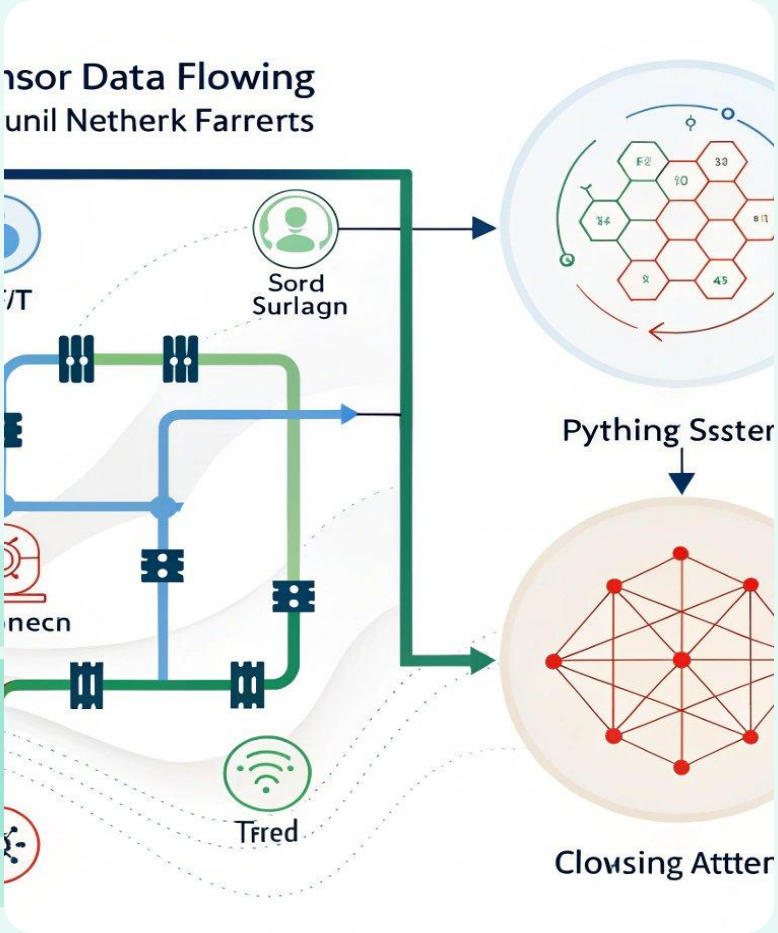
Optional:

Simulated ECLSS subsystem modules for testing and demonstration purposes.

Prototype Software Requirements

Data Collection & Integration

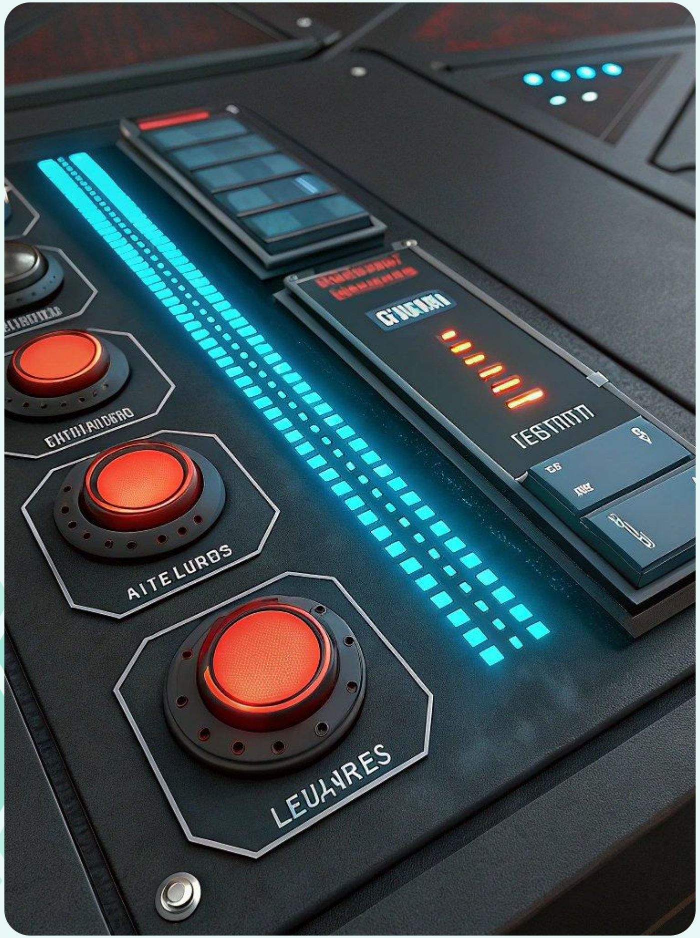
- Software modules for ingesting and synchronizing logs from multiple subsystems and sensors.
- Real-time data streaming and storage (e.g., time-series databases).
- AI/ML Model Development:
  - Libraries for data preprocessing, feature extraction, and model training (e.g., Python, TensorFlow, PyTorch, scikit-learn).
  - Implementation of anomaly detection and predictive maintenance algorithms (e.g., LSTM for time-series, clustering for anomaly detection).



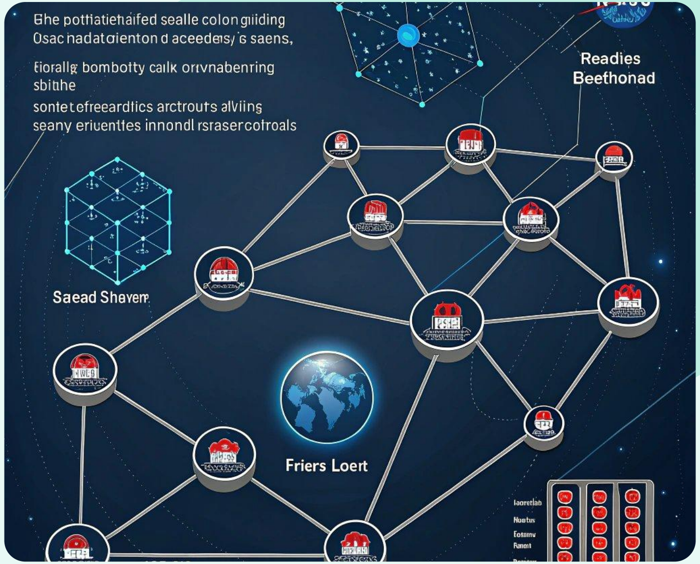
Prototype Software Requirements (Cont.)

User Interface

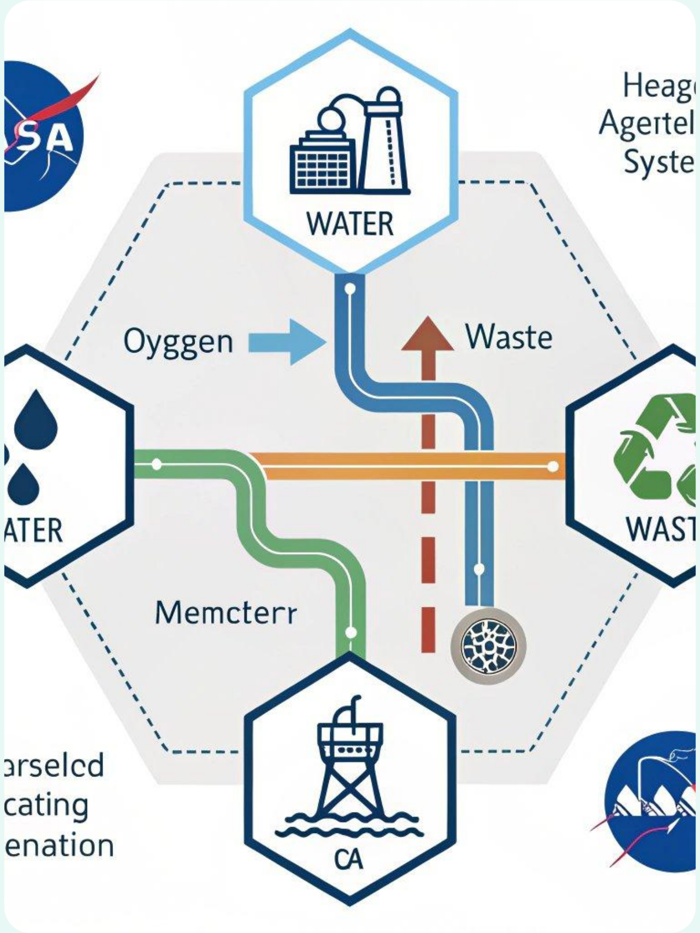
- Dashboard for real-time monitoring, alerts, and visualization of subsystem health.
- Notification system for alarms and maintenance recommendations.
- Simulation & Testing:
  - Tools for generating synthetic data and simulating subsystem behavior if real logs are unavailable.



Constraints



- Data Availability: Access to real ECLSS subsystem logs and sensor data may be limited due to security or operational constraints.
- System Complexity: ECLSS is a nonlinear, multi-level system with complex interdependencies.
- Hardware Limitations: Limited computational resources on prototype hardware may restrict model complexity.
- Safety and Reliability: Predictive models must minimize false positives/negatives to avoid unnecessary interventions or missed failures.
- Integration: Ensuring compatibility with existing NASA systems and data formats.



Training Data Acquisition from ECLSS System and Subsystem LOGS

Data Types Needed:

- Real-time and historical logs from ECLSS subsystems (water, air, oxygen, waste).
- MCA data: concentrations of N2, O2, H2, CO2, CH4, and H2O vapor.
- Sensor data: temperature, humidity, pressure, etc.
- Alarm and event logs for anomaly labeling. Data Collection Methods:
- Direct interface with ECLSS and MCA systems (if available).
- Use of NASA-provided datasets or simulated data for initial development. Data Preprocessing:
- Cleaning, normalization, and feature extraction to ensure data quality and usability. Data Storage:
- Centralized data repository (e.g., data lake or time-series database) for easy access and integration.

TIPS: What If I Can't Get the Logs for Training? What Can I Do

- **Synthetic Data Generation**: Create simulated logs that mimic real system behavior and failure patterns. This allows for model development and testing in the absence of real data.
- **Use of Historical or Public Data**: Leverage any available historical logs or open datasets from similar systems to bootstrap model training.
- **Anomaly Detection with Unlabeled Data**: Employ unsupervised learning techniques (e.g., clustering, autoencoders) to detect anomalies without labeled failure data.
- **System Simulations**: Build or use existing simulations of **ECLSS** subsystems to generate realistic operational data.
- **Expert Knowledge**: Incorporate rules and heuristics from system experts to approximate failure conditions and maintenance needs.

Summary of Benefits

- **Enhanced Safety**: Early detection of potential failures reduces risk to crew and mission.
- **Operational Efficiency**: Optimized maintenance schedules minimize downtime and resource usage.
- **Cost Savings**: Predictive maintenance reduces unnecessary part replacements and labor costs.
- **Scalability**: The approach can be adapted to other **NASA** systems and future missions, supporting the **Artemis** and **Moon to Mars** initiatives.
- **Innovation**: Fosters creative problem-solving and prepares students for **STEM** careers in aerospace and beyond.

Summary Table

Scenario	Astronaut Time	Elapsed Downtime	Net Time Savings vs. Manual
Manual (Standard)	35–55 min	35–55 min	0 (baseline)
AI Predictive	15–25 min	15–25 min	20–30 min
AI + TTE	13–18 min	13–18 min	22–37 min

Key Points

- **AI Predictive Maintenance** typically cuts astronaut time and system downtime by more than **50%**.
- Adding **TTE** shaves off a few more minutes by clarifying urgency and enabling even faster, more focused response.
- **Net Savings:**
  - **Astronaut time saved:** ~20–37 minutes per incident
  - **Elapsed system downtime saved:** ~20–37 minutes per incident