**SOLUTION PLAN EZRA BAKATUBIA**

**AI Approach**

The primary AI technique will be supervised machine learning trained on labeled failure and operational data. This includes classification (healthy vs. degrading vs. critical) and time-series prediction models.

The following models could be considered. Random Forests, being good for tabular, interpretable prediction. XGBoost specifically for high-performance gradient boosting. LSTM, for time series sensor data that depends on temporal trends.

Ensemble methods could also be explored for higher accuracy. The model will continuously predict the failure probability of each monitored component, similar to how engines on Earth-bound aircraft are monitored.

For parts with limited failure data, unsupervised anomaly detection (like Isolation Forest or Autoencoders) will serve as a backup system to catch irregular patterns.

**Data Requirements**

**Input Data Types:**

* Sensor readings (temperature, voltage, vibration, current)
* System status logs (operational mode, error codes, reboots)
* Time-to-failure or failure logs (actual failure events)
* Environmental data (radiation exposure, micrometeoroid impacts)

**Sources:**

* NASA Prognostics Data Repository
* Spacecraft simulation datasets (like NASA’s DASHlink)
* Synthetic data generation for stress testing

Each component is planned to have a set of relevant sensor inputs that need to be processed over time. For example a battery unit might require voltage, internal resistance, and current draw. A motor may need RPM, temperature, and vibration analysis.

**System Architecture / Workflow**

For the input pipeline, the preprocessing includes normalization, resampling of data, and missing value imputation. Feature extraction includes moving averages, standard deviation windows, and FFT transforms.

**Prediction Engine**

The model continuously outputs a risk score between 0–1. Scores are interpreted as:

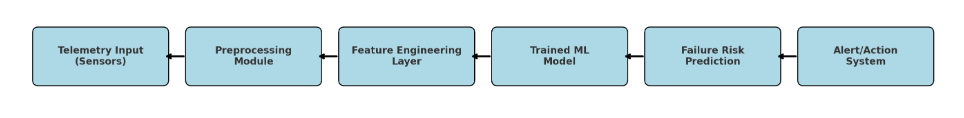
* + 0.0–0.3: Healthy
  + 0.3–0.7: Watch List
  + 0.7–1.0: High Failure Risk

**Output Pipeline**

* System alerts mission control or onboard systems.
* Suggests next actions: isolate part, schedule replacement, shift operations.

We can also add a continuous loop improvement function so after confirmed failures or maintenance events, the system updates its learning to refine predictions in future missions.

**Flow**



**Assumptions**

We’re assuming there’s enough examples of different failures to train the AI right. If not, we’ll just use synthetic or simulated data will be considered acceptable to bootstrap the system. Also, we’re counting on the delay between the spacecraft and Earth not messing things up too much – like, it’ll be smart enough on its own to handle stuff if needed. We’re also assuming the sensors work properly and don’t degrade get worse over time. And just to be clear, this AI isn’t going to run everything by itself, It’s more like a sidekick to the humans controlling everything.

**Limitations and Risks**

One big problem is we don’t have that much real data of equipment damaging in space because it’s not too common. So, we’ll probably need to make up some data through simulations. Another thing is that some AI models, like neural networks, are like black boxes, they work, but it’s hard to explain *why* they made a certain decision, and engineers might not trust that. But even with all that, having a system that can *predict* problems instead of just reacting after something breaks is a huge upgrade. A predictive AI solution presents a powerful improvement over traditional reactive maintenance system.