Testing and Evaluation Plan

Project: AI-Powered Predictive Maintenance for NASA Deep Space Missions

Author: Rubén Darío Valenzuela Alvarado

Course: ITAI 2372 – Artificial Intelligence Applications

Track: Conceptual Design Track

# 1. Testing Methodology

Although this is a conceptual project, the testing strategy is designed to reflect how the proposed AI system would perform under real mission conditions using simulated telemetry data.  
  
We will focus on three primary methods:  
- Simulated Fault Injection: Create a dataset simulating normal operations and known anomalies (e.g., power loss, overheating, communication failure).  
- Synthetic Telemetry Streaming: Emulate real-time sensor input from spacecraft systems (temperature, vibration, voltage, signal latency).  
- Scenario-Based Evaluation: Test the AI system’s response to long-duration mission stressors (component degradation, radiation interference, solar events).

# 2. Test Scenarios

|  |  |  |
| --- | --- | --- |
| Scenario | Description | Expected System Behavior |
| Overheating | Simulate a gradual rise in temperature in propulsion systems | Early detection + send alert to mission control |
| Vibration spike | Emulate hardware shaking caused by micrometeorites | Classify anomaly + suggest diagnostic check |
| Battery degradation | Simulate power storage decay over time | Forecast failure timeline + schedule maintenance |
| Sensor error | Introduce false sensor readings | Recognize inconsistency + request data validation |

# 3. Evaluation Metrics

|  |  |
| --- | --- |
| Metric | Purpose |
| Precision | Accuracy in identifying true anomalies |
| Recall | Ability to detect all relevant issues |
| Latency | Time taken to detect and respond to issues |
| False Positives | Avoiding unnecessary alerts |
| System Uptime | Maximize mission continuity |

# 4. Simulation Tools and Data

- Python-based Simulations using NumPy and Pandas  
- MATLAB Simulink for sensor modeling (optional)  
- NASA’s open telemetry datasets (e.g., Mars Science Laboratory)  
- Custom JSON/CSV logs to simulate data streaming

# 5. Success Criteria

- At least 90% detection accuracy across anomaly types  
- Low false positive rate (below 5%)  
- Maintenance suggestions generated within 5 seconds of issue detection  
- System handles continuous simulated data for 24+ hours without crash or performance drop