

PrognosAI: AI-Driven Predictive Maintenance System Using Time-Series Sensor Data

Project Objective

To design and develop an AI-based predictive maintenance system capable of estimating the Remaining Useful Life (RUL) of industrial machinery using multivariate time-series sensor data. The system will be prototyped using the NASA CMAPSS dataset and structured to generalize across similar domains such as turbines, pumps, and motors. The objective is to enable timely maintenance decisions, minimize unplanned downtime, and optimize asset utilization by applying deep learning techniques like LSTM for sequential pattern recognition and failure prediction.

Project Workflow

1. Data Ingestion

- Load and preprocess the CMAPSS sensor dataset (cycle-wise engine data).

2. Feature Engineering

- Create rolling window sequences and compute Remaining Useful Life (RUL) targets.

3. Model Training

- Train a time-series model (e.g., LSTM or GRU) to predict RUL from sensor sequences.

4. Model Evaluation

- Evaluate performance using RMSE and compare predicted RUL vs actual RUL.

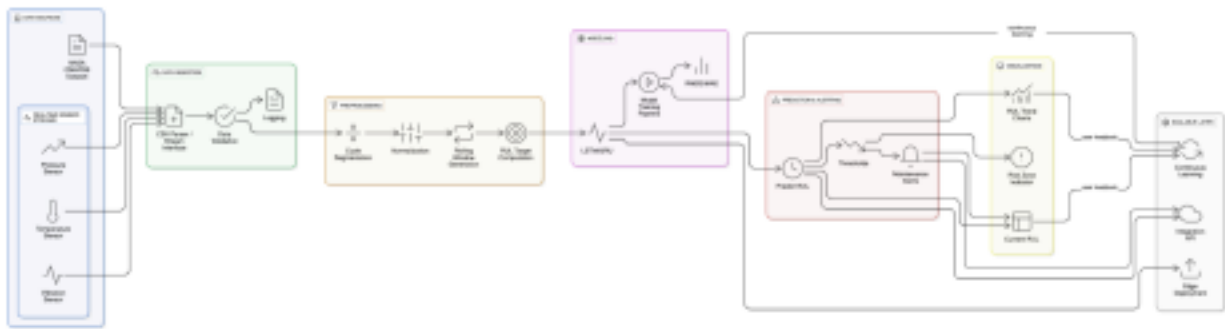
5. Risk Thresholding

- Define thresholds to trigger maintenance alerts based on predicted RUL.

6. Visualization & Output

- Present results via charts and dashboards showing RUL trends and alert zones.

Architecture Diagram



eraser

Tech Stack

- **Python** – Core programming language
- **Pandas, NumPy** – Data processing
- **Matplotlib, Seaborn** – Visualization
- **TensorFlow / Keras** – LSTM model training
- **Scikit-learn** – Metrics & preprocessing
- **Streamlit / Flask** – Dashboard or API interface
- **Docker** – Optional deployment
- **NASA CMAPSS Dataset** – Source data

Milestone 1: Data Preparation & Feature Engineering

- **Objective:** Successfully load, preprocess, and prepare the CMAPSS dataset for model training, including the creation of appropriate features and RUL targets.
- **Deliverables:**
 - Cleaned and preprocessed CMAPSS sensor data.
 - Python scripts for data loading and preprocessing.
 - Generated rolling window sequences.
 - Computed RUL targets for all engine cycles.
- **Evaluation:**
 - Verification of data integrity and absence of missing values.
 - Correctness of rolling window sequence generation.
 - Accuracy of RUL target calculations.
 - Documentation of data preparation steps.

Milestone 2: Model Development & Training

- **Objective:** Develop and train a time-series deep learning model (LSTM/GRU) capable of learning patterns from sensor data to predict RUL.
- **Deliverables:**

- Implemented LSTM or GRU model architecture.
- Trained model weights.
- Training loss and validation loss curves.
- Code for model definition, training, and saving.
- **Evaluation:**
 - Convergence of training process (loss reduction).
 - Initial performance on validation set (e.g., visual inspection of predicted vs. actual RUL).
 - Proper implementation of the deep learning model.

Milestone 3: Model Evaluation & Performance Assessment

- **Objective:** Rigorously evaluate the trained model's performance using defined metrics and analyze its predictive accuracy.
- **Deliverables:**
 - Calculated RMSE scores for the test set.
 - Plots comparing predicted RUL against actual RUL.
 - Analysis of model biases and errors.
 - Detailed evaluation report.
- **Evaluation:**
 - Achieve an acceptable RMSE score (e.g., below a predefined threshold).
 - Visual consistency between predicted and actual RUL plots.
 - Identification and understanding of model limitations.

Milestone 4: Risk Thresholding & Alert System

- **Objective:** Define and implement a mechanism to translate RUL predictions into actionable maintenance alerts.
- **Deliverables:**
 - Defined RUL thresholds for different alert levels (e.g., warning, critical).
 - Logic for triggering maintenance alerts based on predicted RUL.
 - Examples of triggered alerts.
- **Evaluation:**
 - Effectiveness of thresholds in identifying potential failures early.
 - Clarity and accuracy of alert triggers.
 - Practical applicability of the alerting mechanism.

Milestone 5: Visualization & Dashboard Development

- **Objective:** Create interactive visualizations and a dashboard to present RUL trends, predictions, and alerts.
- **Deliverables:**
 - Interactive charts displaying RUL trends over time.

- Dashboards showing current RUL predictions and alert zones.
- User-friendly interface (e.g., Streamlit/Flask application).

- **Evaluation:**

- Clarity and informativeness of visualizations.
- Responsiveness and usability of the dashboard.
- Ability to convey critical insights about asset health and potential risks.