

AI-PrognosAI Predictive Maintenance System

Milestone 3: Evaluation & Performance Assessment

Project Overview

This milestone continues the development of a hybrid deep learning system for Remaining Useful Life (RUL) prediction using sensor time-series data. Building on previous milestones, the focus here is on model evaluation, metric analysis, and interpretation using test data.

Setup & Dependencies

Python 3.x

Libraries: numpy, pandas, matplotlib, seaborn, scikit-learn, tensorflow, keras

Data source: sensordata.csv (or simulated data if not present).

Step-by-Step Documentation

1. Data Preparation & Rolling Window Feature Engineering

Normalized sensor data using StandardScaler.

Created rolling window sequences (WINDOWSIZE=30, STEPSIZE=5) for time-series feature extraction.

Computed RUL targets, with linear decrease capped at non-negative minimum.

Split dataset: 70% train, 15% validation, 15% test.

2. Model Development (Summary from Previous Milestone)

Built a hybrid sequential model: Conv1D layers, BatchNormalization, Bidirectional LSTM layers, Dense layers for regression.

Architecture:

2 Conv1D layers

2 BatchNormalization + Dropout layers

2 Bidirectional LSTM layers (64, 32 units)

Output: Dense regression head.

3. Training & Model Checkpointing

Optimizer: Adam (learning rate 0.001), loss: MSE, metrics: MAE, MSE.

Training with callbacks: EarlyStopping (patience=15), ReduceLROnPlateau, ModelCheckpoint.

Epochs: 100, Batch size: 64.

Visualized training history (Loss, MAE over epochs).

4. Evaluation & Performance Metrics

Predictions generated on the test set.

Key metrics reported:

RMSE: 3.70

MAE: 3.15

R2: 0.98

Model explains 98.37% of variance in RUL.

Interpretation statements included for metric context (e.g., "Excellent performance if RMSE < 10 cycles").

5. Visualization & Interpretation

Plotted actual vs predicted RUL (scatter plot, time series for first 200 samples).

Illustrated error distribution (histogram of prediction errors, mean/std annotated).

Comments describe each visualization, supporting reproducibility for others.

Usage Example

```
# Load dataset & preprocess
df = pd.read_csv('sensordata.csv')
# Follow milestone script for rolling window, normalization, RUL target
calculation

# Model prediction
ypred = model.predict(X_test).flatten()

# Metric calculation
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
```

Refer to the notebook source for end-to-end running details.

Result Summary

Milestone 3 completes the evaluation and model interpretation phase, confirming strong

predictive performance and reliability with insights provided by visualization and metrics analysis.